ITERATIVE METHODS FOR SOLVING LARGE-SCALE LINEAR PROGRAMS

By

Gollakota Surya Kumar

A DISSERTATION PRESENTED TO THE GRADUATE COUNCIL OF
THE UNIVERSITY OF FLORIDA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1979

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to Professor Gary Koehler, my dissertation chairman, for the excellent guidance, the constant encouragement and the support he has provided me during my career at the University of Florida. Despite his busy schedule he was always available at a moment's notice.

I also wish to thank Professor Antal Majthay whose depth of knowledge not only in the area of operations research but on a wide range of subjects will always remain awe-inspiring. I am also grateful to Professors Ira Horowitz and Henry Tosi for providing financial assistance during my stay at the University of Florida. I also wish to thank Ms. Kathy J. Combs for her superb typing of the manuscript.

Last but not the least this work would not have been possible but for the patience, understanding and encouragement of my wife Rekha.

TABLE OF CONTENTS

Abstract of Dissertation Presented to the Graduate Council
of the University of Florida
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

ITERATIVE METHODS FOR SOLVING
LARGE-SCALE LINEAR PROGRAMS

By

Gollakota Surya Kumar

August 1979

Chairman: Gary J. Koehler
Major Department: Management

The simplex method for solving linear programs has achieved its overwhelming success over the years due to the advances during the last three decades on extending the effectiveness of the basic algorithm introduced by Dantzig in 1951. However the largest problem that can be economically solved by this method is dictated by the size of the basis. The limitation being the ability to maintain a sparse yet accurate representation of the basis inverse. Commercial computer codes using the recent developments of the simplex method are capable of solving mathematical program systems of the order of 10,000 rows. However, larger linear programs do arise in practice. The reason why they are not being set up is because none of the production LP codes can solve such large problems in a timely and cost effective fashion. For such large

problems iterative techniques that do not require a basis may be an attractive alternative.

We review two classes of iterative techniques--relaxation methods and subgradient optimization. Solving a linear program can be shown to be identical to finding a point of a polyhedron P, the polyhedron being formed from primal and dual constraints and an additional constraint using strong duality. In relaxation methods we successively find points of relaxed forms $P \subset P^n$. That is, at iteration n we find an $x^n \in P^n$ such that $P \subset P^n$. We assume $P \neq \emptyset$. The problem is solved when the sequence converges to a point $x^* \in P$.

Subgradient methods are an alternative class of iterative techniques for solving these problems. In subgradient methods we minimize a convex though not necessarily differentiable function $f(\cdot)$ by the iterative scheme $x^{n+1} = x^n - t^n u^n$, where $t^n > 0$ is a a scalar obtained by the approximate minimization of $f(x^n - t^n u^n)$, and u^n is a subgradient of f at x^n. When we use the minimization method for finding a point of a polyhedron P, the function f is so defined that if x^* is a minimum of f then $x^* \in P$.

These iterative techniques work with original data; thus advantages of supersparsity can be fully realized and the program run in core. Also they are self correcting and knowledge of a vector close to the solution can be used to advantage.

We show that a generalized form of Merzlyakov's relaxation procedure subsumes most of the recent subgradient procedures when the objective is to find a point of a polyhedron. A new algorithm

using relaxation method is presented. This algorithm was coded to explore values for the relaxation parameters used in the algorithm.

CHAPTER 1

THE PROBLEM: WHY ITERATIVE TECHNIQUES?

Mathematical programming has achieved its present popularity in a large measure due to the success of the simplex method for solving linear programs published by Dantzig [6] in 1951. Over the years considerable effort has been expended towards exploiting and extending the effectiveness of this procedure. We will briefly outline some of the improvements.

Most practical linear programs have a sparse matrix [2] and it is necessary to exploit this sparseness to reduce memory and time requirements and to maintain accurate information. The first such procedure was the revised-simplex procedure. This procedure operates only on original problem data and requires an inverse of the current basis. If there are m constraints and n variables (including slacks, surplus and artificials) then the regular simplex procedure requires

$$(1.1) \qquad (m + 1)(n + 1)$$

storage locations. There are m + 1 rows (including the objective and n + 1 columns (including the right-hand side). In the revised-simplex procedure we need

$$(1.2) \qquad p(m + 1)(n + 1) + m^2$$

where p is the problem density. (1.1) is much larger than (1.2) if p is close to zero.

It was also discovered that if the inverse matrix were stored in product form [29] an additional decrease in storage requirements could be realized. However, in product form, the basis inverse grows in size at each iteration so that eventually we must either run out of room or scrap the current form of the basis inverse and reinvert the current basis. In practice this happens quite often. Using product form inversion, the storage requirements are

(1.3) $p(m + 1)(n + 1) + qmr$

where q is the average density of the eta vectors needed in the product form inverse and r is the average number of eta vectors. Notice, though, that to gain the decrease in (1.3) over (1.2) we must spend time in the reinversion step.

It was soon realized that there are many ways of forming a product form of a basis matrix. For instance, the matrix could first be decomposed into the product of two triangular matrices. The eta vectors for these are sparse and easily formed. In addition, within a class of procedures for viewing the original basis, one can choose the row-column pivot order to minimize the creation of new members -- i.e. one can choose the order in forming the eta vectors to minimize the density of the resulting basis inverse representation [20]. A related issue has to do with updating the currect basis inverse with a new eta vector. One can perform this operation with a minimal buildup in storage requirements [14].

Kalan [22] pointed out that of the $p(m + 1)(n + 1)$ elements of a typical LP problem, most are identical (usually most are $\pm$ 1s).

Kalan suggested storing only the unique numbers of an LP problem and using pointers to store the problem structure. In so doing one stores fewer elements than exist in the problem. The new sparseness is termed supersparsity. Kalan noted that not only can original data be stored this way but also new data (formed in the basis inverse) can be stored in this fashion.

Another useful idea -- scaling of data -- has numerical stability for its motivation. Generally for real problems the coefficients of matrix may take on a wide range of values. This can cause numerical difficulties -- for example in inversion -- since the computer must round off fractions. To reduce the effect of this problem, the matrix entries are scaled before starting the problem.

There are a number of ways in which to obtain a starting basis: from scratch using an "artificial basis," using a CRASHed basis forcing some of the variables into the basis, or based on information about the problem. The last approach is especially useful in solving a series of problems with only minor variations in data.

Candidate selection can play an important part in the efficiency of an LP algorithm. There are a number of ways to select the entering variable -- "multiple pricing" is one of them which is especially useful for large problems. In this approach we typically divide the variables into segments and pick the segments sequentially. In the chosen segment we select the g most negative reduced cost variables as candidates for pivoting. We concentrate on these g

non-basic variables for the remainder of some h iterations. And then repeat the procedure with the next segment. A more recent development concerning pivot selection is due to Paula Harris [17]. The reduced costs here are weighted prior to selection. The weights are chosen with respect to a fixed set of non-basic variables. The idea is that in the traditional most negative reduced cost approach we are trying to find out the best candidate but on a local basis -- locally we move in the direction of steepest ascent of the objective function. Such a selection may not serve global interests. By constantly adjusting the pricing by weights derived from a fixed reference point, the global interests of the problem are hopefully maintained. Experimental results corroborate this expectation and show it to be a very effective procedure.

When there is degeneracy in a linear programming problem, there exists the possibility of a set of basic feasible solutions occurring repeatedly -- called "cycling." Bland [3] has introduced several new methods to avoid cycling -- the simplest of which is to select as entering variable one with the smallest subscript among all candidates to enter basis. If there is a tie for leaving variable, select the one with the smallest basic subscript.

Parallel with efforts to increase the size of problems that can be tackled by the simplex method have been efforts to increase its capabilities for problems having special structure. The computing effort using the simplex method depends primarily on the number of rows; so it is worthwhile considering the possibility of

taking care of some constraints without increasing the size of the basis. Generalized Upper Bounding (GUB) technique introduced by Dantzig and Van Slyke [9] achieves this for contraints of the form

$$(1.4) \qquad \sum_j x_j = b_i$$

with the restriction that the same variable does not occur in more than one GUB constraint. Decomposition technique is another method of extending the capability of problem solving by breaking it into smaller subproblems and using a master program iteratively to tie them together. Unfortunately the convergence of this method is rather slow and in addition one is unable to compute bounds on the value of the objective function at each iteration, unless each of the subproblems is bounded.

Despite the far reaching advances made during the last three decades in extending the capability of the simplex method, some of which have been mentioned above, the basis inverse still remains the limiting factor dictating the size of the problem that could be tackled. Even when we use variants like GUB and decomposition techniques, the size of the basis is still the bottleneck. In the case of GUB it is the basis of the master problem consisting of other than GUB constraints and with decomposition it is the size of the basis of the largest of subproblems. Thus this basis inverse is the heart of all that is bad in simplex method related techniques of solving an LP.

Commercial computer codes using above developments of simplex method are capable of solving mathematical programming systems of the order of 10,000 rows. However, useful linear programs of such

magnitude do arise in practice for example in food, oil and chemical related industries. The reason why larger ones are not being set up and solved is that no production LP code has been developed to solve such large problems in a timely and cost effective fashion. This is not to say that there are not large linear programming production codes. For example, IBM's MPSX/370, CDC's APEX, Honeywell's LP 6000 X, Techtronic's MPS3, and others are reliable and available at reasonable rates. However, one would not seriously consider using one of these to solve a linear program of 10,000 or more rows on a routine basis unless the problem possessed some nice structure or the results of the problem solution had a high economic value.

Dantzig et. al. [8] in their justification for the setting up of a system optimization lab for solving large scale models have noted:

> Society would benefit greatly if certain total systems can be modeled and successfully solved. For example, crude economic planning models of many developing countries indicate a potential growth rate of 10 to 15% per year. To implement such a growth (aside from political differences) requires a carefully worked out detailed model and the availability of computer programs that can solve the resulting large-scale systems. The world is currently faced with difficult problems related to population growth, availability of natural resources, ecological evaluations and control, urban redesign, design of large scale engineering systems (e.g. atomic energy and recycling systems), and the modeling of man's physiological system for the purpose of diagnosis and treatment. These problems are complex, and urgent and can only be solved if viewed as total systems. If not, then only patchwork piecemeal solutions will be developed (as it has been in the past) and the world will continue to be plagued by one crisis after another caused by poor planning techniques.

The bottleneck in solving such large unstructured LPs is the inability to maintain a sparse yet accurate representation of the basis inverse. When the simplex method is not computationally feasible for such large problems, iterative techniques that do not require a basis inverse may be an attractive alternative. In this dissertation we concentrate on iterative procedures for solving linear programs.

We discuss iterative techniques under the headings (a) relaxation methods and (b) subgradient methods. These methods are insensitive to the number of constraints. In fact an increase in the number of constraints for the same problem improves the convergence of the relaxation methods. Unlike the simplex method, iterative techniques are self-correcting. We always use original data thus advantages of supersparsity [22] can be fully realized and the program run in main memory. Finally knowledge of a vector close to the solution can be used to advantage. This is very useful when a sequence of problems has to be solved with only slightly varying data.

As an aside to the main thrust of this dissertation, we note that iterative methods are more attractive for solving certain large Markov decision processes and also for problems having a Leontief Substitution constraint set [23, 25]-both of these problems are an important special class of LPs.

CHAPTER 2

BACKGROUND RESULTS AND TOOLS

Introduction

Motivation

In this chapter we will review various iterative techniques for finding a point of a polyhedron. We are interested in finding a point of a polyhedron because a wide variety of important mathematical programs can be reduced to such a problem. For example, solving a linear program could be reduced to finding a point of a polyhedron using primal and dual constraints along with an additional constraint implying strong duality. To elucidate this further, consider the following primal-dual linear programming pair:

(2.1)

Primal	Dual
Max $c'x$	Min $v'b$
s.t. $Ax \leqq b$	s.t. $v'A \geqq c$
$x \geqq 0$	$v \geqq 0$

Here A is m x n; c is n x 1 and b is m x 1. v' stands for transpose of v. Problem 2.1 can be restated as

$$(2.2) \quad \begin{aligned} -Ax + b &\geqq 0 \\ v'A - c &\geqq 0 \\ v'b - c'x &\geqq 0 \\ v \geqq 0,\ x &\geqq 0 \end{aligned}$$

Let $P = \{\binom{x}{v}: \binom{x}{v}$ satisfies 2.2$\}$ be the set of solutions to 2.2.

$\binom{x}{v} \varepsilon P$ implies that x solves the Primal problem in 2.1 and v solves the dual problem.

Linear programs of the form

$$z^* = \max_{x} \min_{i=1,\dots,m} (a_i'x + b_i)$$

relate directly to the problem of finding a point of a polyhedron P where

$$P = \{x \varepsilon R^k: a_i'x + b_i \geq z^*, \ i = 1, \dots, m\}$$

Such problems arise, for example, in large-scale transportation problems and decomposition problems [27]. We will develop such a formulation for the transportation problem.

Given k origins with stocks $a_\ell > 0$ of a single commodity, and m destinations with demands $d_j > 0$ the transportation problem involves determining a shipping schedule that meets all demand and supply contraints at minimum cost. Let

$z_{\ell j}$ be the amount shipped from ℓ to j

$c_{\ell j}$ cost of shipping one unit from ℓ to j.

The problem is

$$\text{Min} \sum_{\ell=1}^{k} \sum_{j=1}^{m} c_{\ell j} z_{\ell j}$$

subject to

$$\sum_{j=1}^{m} z_{\ell j} = a_\ell \ , \ \ell = 1, \dots, k$$

$$\sum_{\ell=1}^{k} z_{\ell j} = d_j \ , \ j = 1, \dots, m$$

$$z_{\ell j} \geqq 0$$

Without loss of generality, we assume

$$\sum_{\ell=1}^{k} a_\ell = \sum_{j=1}^{m} d_j$$

If m and k are too large to be handled by conventional network methods [4], then the decomposition principle can be used. Let

$$\{(z^i_{\ell j}: \ell = 1, \ldots, n; \ j = 1, \ldots, m): i \in I\}$$

be a list of extreme points of

$$\sum_{\ell=1}^{k} z_{\ell j} = d_j, \ j = 1, \ldots, m$$

$$z_{\ell j} \geqq 0$$

Then the transportation problem is equivalent to the master problem

$$\text{Min} \sum_{\ell=1}^{k} \sum_{j=1}^{m} c_{\ell j} \sum_{i \in I} z^i_{\ell j} \lambda^i$$

subject to

$$a_\ell - \sum_{i \in I} \sum_{j=1}^{m} z^i_{\ell j} \lambda^i = 0, \ \ell = 1, \ldots, k$$

$$\sum_{i \in I} \lambda^i = 1$$

$$\lambda^i \geqq 0.$$

Let

$$b_i = \sum_{\ell=1}^{k} \sum_{j=1}^{m} c_{\ell j} z^i_{\ell j}$$

$$\bar{a}^i_\ell = a^i_\ell - \sum_{j=1}^{m} z^i_{\ell j}$$

The master problem becomes

$$\text{Min} \sum_{i \in I} b^i \lambda^i$$

subject to

$$- \sum_{i \varepsilon I} \bar{a}^i_\ell \lambda^i = 0, \ \ell = 1, \ldots, k$$

$$\sum_{i \varepsilon I} \lambda = 1$$

$$\lambda^i \geqq 0$$

The dual to the above problem is

$$\text{Max } w$$

subject to

$$-\sum_{\ell=1}^{k} a^i_\ell x_\ell + w \leqq b^i$$

$$w \leqq b^i + a'_i x$$

where

$$a_i = (\bar{a}^i_1, \bar{a}^i_2, \ldots, \bar{a}^i_k) \ \varepsilon \ R^k$$

This problem is equivalent to

$$\underset{x \varepsilon R^k}{\text{Max}} \ \underset{i \varepsilon I}{\text{Min}} \ \{a'_i x + b_i\}$$

Task

Two of the methods of finding a point of a polyhedron are the Fourier-Motzkin Elimination Method [7] and a Phase I simplex method. The Elimination Method reduces the number of variables by one in each cycle but may greatly increase the number of inequalities in the remaining variables. For larger problems this procedure is impractical [26] -- since the number of inequalities grows at a rapid rate. The Phase I simplex procedure can also be used to find a point of a polyhedron. For very large problems the simplex method breaks down due to a practical limitation on the size of the matrix

that can be inverted as explained in the last chapter. It is for problems of this nature that iterative methods may be an attractive alternative. We discuss iterative techniques under two classes -- relaxation methods and minimization methods.

In relaxation methods, at each iteration we attempt to find a point of a relaxed form P^n of P. That is at iteration n we find an $x^n \varepsilon P^n$ such that $P \subset P^n$. We look for the property that the sequence of points thus generated is pointwise closer to P. Minimization methods attempt to solve the problem of minimizing a general, maybe non-differentiable, convex function $f(\cdot)$. The sequence generated generally has the property that the successive iterates are closer to the level set of solutions in distance. x^* is one such point. The value of the objective function need not necessarily improve at each iteration. When we use the minimization method for finding a point of P, the function f is so defined that if x^* is a minimum of f then $x^* \varepsilon P$.

We will now review the relaxation methods. As was mentioned earlier, we find at each iteration n, a point $x^n \varepsilon P^n$ where P^n is a relaxed form of P. That is $P \subset P^n$. We continue finding points successively in this manner till we can find an $x^* \varepsilon P$. At least one such point exists by our assumption that $P \neq \emptyset$.

Definitions

We need a few definitions before some of the methods can be described. Let

$$\ell_i(x) = a_i'x + b_i$$

where $i \in I$, $I \neq \emptyset$ and finite, $x \in R^k$, $a_i \in R^k$ and $b_i \in R$. Without loss of generality assume $\|a_i\| = 1$ for all i, where $\|\cdot\|$ is the Euclidean norm. H_i given by

$$H_i = \{x \in R^k \colon \ell_i(x) \geq 0\}$$

is the <u>halfspace</u> associated with $i \in I$. P given by

$P = \bigcap_{i \in I} H_i = \{x \in R^k \colon \ell_i(x) \geq 0,\ j \in I\}$ is the <u>polyhedron</u>

of interest. E_i given by

$$E_i = \{x \in R^k \colon \ell_i(x) = 0\}$$

is the <u>plane</u> associated with halfspace H_i . $d(x, H_i)$ given by

$$d(x, H_i) = \inf_{z \in H_i} \|x-z\|$$

is the Euclidean <u>distance</u> from $x \in R^k$ to halfspace H_i. $d_p(x)$ given by

$$d_p(x) = \max_{i \in I} d(x, H_i)$$

is the <u>maximal distance</u> from x to P. $B_r(x)$ given by

$$B_r(x)\ \{y \in R^k \colon \|x-y\| \leq r\}$$

is the <u>ball</u> with radius r and center x. $S_r(x)$ given by

$$S_r(x) = \{y \in R^k \colon \|x-y\| = r\}$$

is the k-1 dimensional <u>sphere</u> of radius r and center x

Typical Relaxation Method

Procedure

A typical relaxation method for finding a point of P is

(1) Pick $x^o \in R^k$ arbitrarily

(2) If $x^n \varepsilon P$ stop, otherwise let i* be the most isolated halfspace, i.e.,

$$-a_{i^*}' x^n - b_{i^*} \geq -a_i' x^n - b_i, \text{ for all } i \varepsilon I.$$

We obtain the sequence $\{x^n\}$ recursively as follows:

$$x^{n+1} = x^n + \lambda^n (t^n - x^n)$$

where $t^n = P_{E_{i^*}}(x^n)$ is the projection of x^n on E_{i^*}. λ^n is called the relaxation parameter at iteration n. Generally λ^n is constant across n = 1, 2, ...

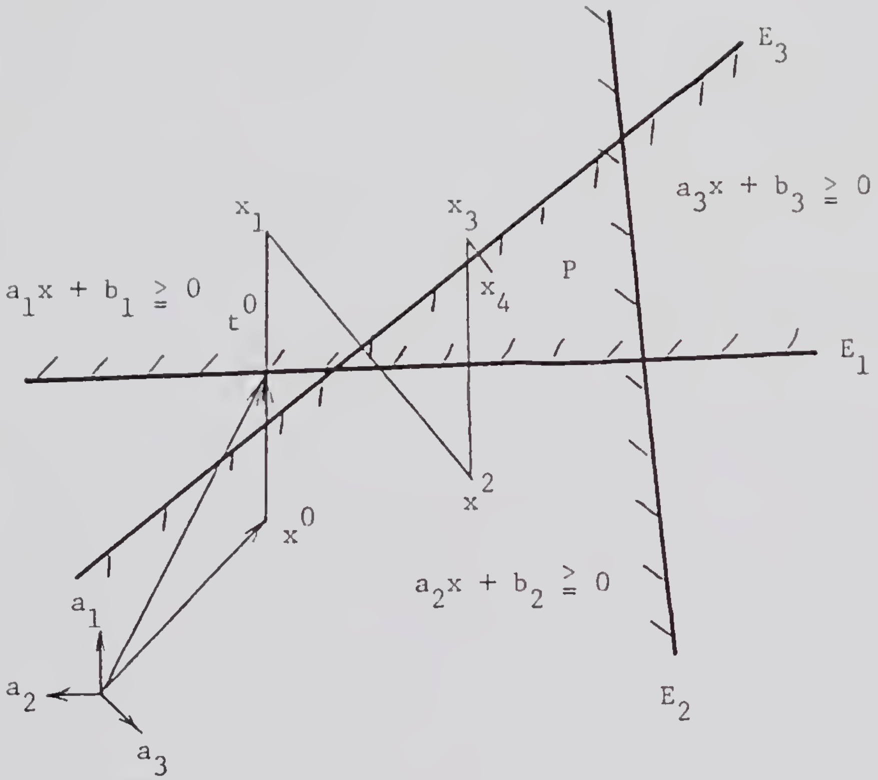
Variations

If $\lambda^n = 1$ for all n, we call the procedure a projection method, and when $\lambda^n = 2$ for all n, we call it a reflection method. See Figure 1 for an example of the relaxation procedure. In this example we use $\lambda^n = \lambda = 2$, for all n. At x°, E_1 is the plane corresponding to the most violated halfspace. We reflect in E_1 to get x^1. At x^1 we reflect in E_3 to get x^2. If we continue the procedure we are in P in four iterations.

Convergence

Fejer-monotone sequence

We are ultimately interested in finding a point of P or at least finding a sequence of points which converge to a point of P. If the sequence terminates with a point of P (as in Figure 1) we have accomplished our goal. If the sequence does not terminate, then we wish to know whether it converges to a point of P. A well known result concerning "Fejér-monotone" sequences will be used to shed light on this question.



E_3
$a_3x + b_3 \geqq 0$
x_1
x_3
P
$a_1x + b_1 \geqq 0$
x_4
t^0
E_1
x^2
x^0
$a_2x + b_2 \geqq 0$
a_1
a_2
E_2
a_3

FIGURE 1

A sequence of points $\{x^n\}$ in R^k is said to be Fejer-monotone [13, 31] with respect to the set P if

(i) $x^{n+1} \neq x^n$ and

(ii) $||x^{n+1} - z|| \leq ||x^n - z||$ for all $z \in P$ and for all n.

<u>Motzkin and Schoenberg</u>

A well known result in analysis is that if $\{s^n\}$ is monotone, then $\{s^n\}$ converges if and only if it is bounded. Hence we see that a Fejer-monotone sequence always converges in distance. However, we need to relate this convergence to the convergence of sequence $\{x^n\}$. Before we relate a theorem on convergence of Fejér-monotone sequence of points, we need a few more definitions.

Given a set $K \subset R^k$, the <u>affine hull</u> of K, denoted by $M(K)$, is

$$M(K) = \{x \in R^k: x = \sum_{i\epsilon L} \lambda^i x^i,\ x^i \in K,$$

$$\sum_{i\epsilon L} \lambda^i = 1,\ L \text{ finite},\ \lambda^i \in R\}.$$

For example, the affine hull of a point is the point itself and affine hull of a line segment is a line containing the segment. The affine hull of a triangle in R^3 is the plane containing the triangle. A set P is <u>affine</u> if $P = M(P)$.

Let M be affine in R^k, x a point of M and $r \in R^+$ be a positive real number. Then we define $S_r(M,x)$ as the <u>spherical surface with axis M, center x and radius r</u> by

$$S_r(M,x) = x + ((M - x)^{\perp} \cap \{z \in R^k: ||z|| = r\}) \tag{2.3}$$

Figure 2 shows the construction of a spherical surface. Given a point of affine set M, M - x is the translation of M by x. $(M - x)^{\perp}$ is orthogonal to M - x. $S_r(0)$ is a spherical surface with radius r,

which intersects $(M - x)^{\perp}$ at two points. These two points are translated back by x to give $S_r(M, x)$.

(2.4) Theorem (Motzkin and Schoenberg)

Let the infinite sequence $\{x^n\}$ be Fejér-monotone with respect to the set P, then

Case (a): If P has an interior, the sequence converges to a point.

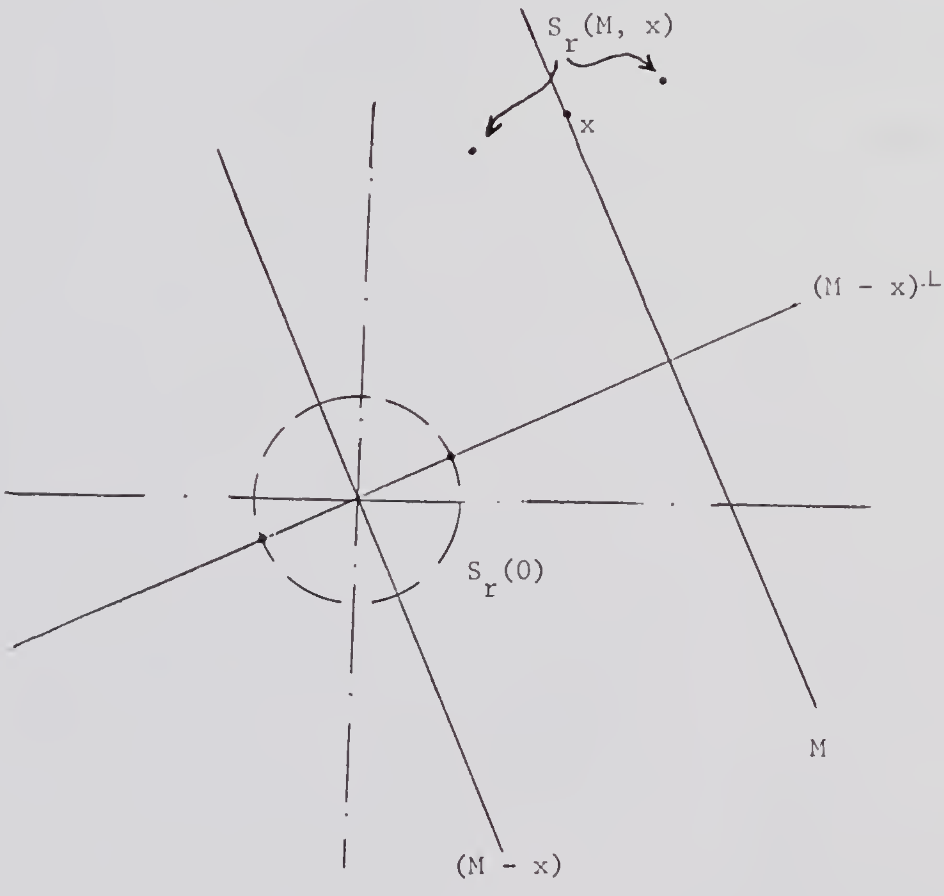


FIGURE 2

Case (b): If P does not have an interior, then the sequence can be convergent to a point or the set of its limit points may lie on a spherical surface with axix $\mathcal{M}(P)$.

See Figure 3 for an illustration. In case (a) P has an interior and the sequence $x^0, x^1, x^2, x^3, \ldots$ converges to a point x^* of P. In case (b) P does not have an interior and the Fejér-monotone sequence $x^0, x^1, x^2, x^3, \ldots$ results in the two limit points of the sequence lying on a spherical surface with axis $\mathcal{M}(P)$ and center y. y being the projection of x^1 on P

Relaxation Procedures

Agmon

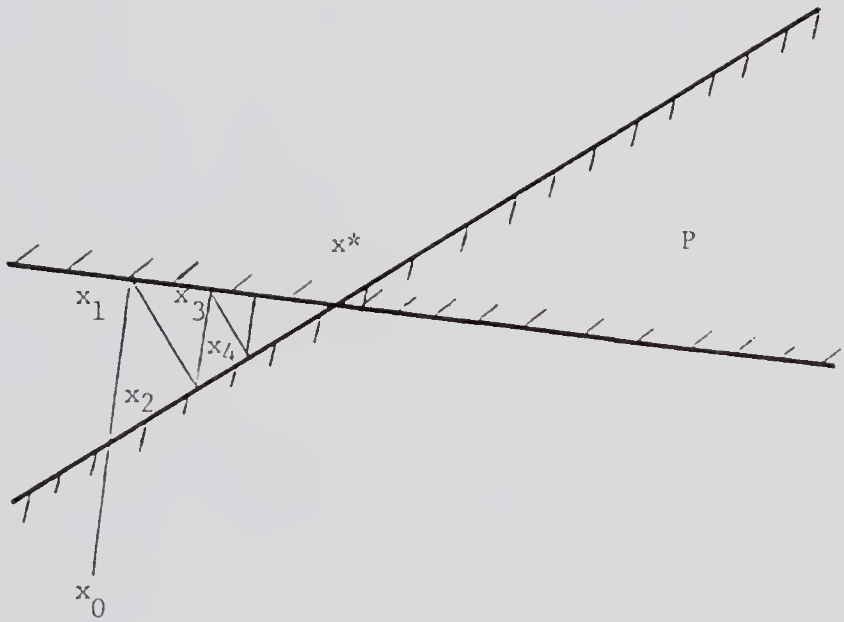
We are now in a position to start our survey of relaxation procedures. One of the earliest published methods is the projection method due to Agmon [1]. Here we specify an initial vector $x^0 \varepsilon R^k$ (chosen arbitrarily) and set $\lambda^n = 1$ for all n. At iteration n if $x^n \varepsilon$ P stop. Otherwise we find a new point x^{n+1} as follows:

$$x^{n+1} = x^n + \lambda^n (t^n - x^n)$$

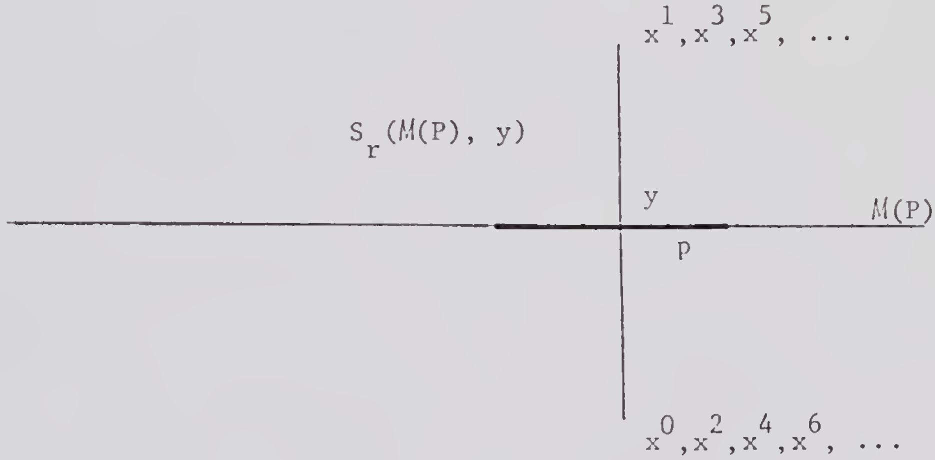
$$= x^n - \frac{(a'_{i_n} x^n + b_{i_n})\, a_{i_n}}{a'_{i_n} a_{i_n}}$$

$$= x^n - (a'_{i_n} x^n + b_{i_n})\, a_{i_n}$$

since $\| a_{i_n} \| = 1$. Here i_n represents a most violated contraint of P at x^n.



Case (a)



Case (b)

FIGURE 3

This procedure yields a sequence which converges to a point of P. In the Agmon procedure

$$P^n = \{x: a'_{i_n} x + b_{i_n} \geqq 0\}$$

That is, P^n is the closed halfspace defined by the constraint i_n and $P \subset P^n$. In keeping with our notation, here P^n is a relaxed form of P.

(2.5) Theorem (Agmon)

Let $P = \bigcap_{i \in I} H_i$, $P \neq \phi$, $I \neq \phi$ and finite, be the solution set of a consistent system of linear inequalities, and let $\{x^n\}$ be the sequence defined by the projection method, that is, $\lambda^n = 1$ for all n. Then either the process terminates after a finite number of steps or the sequence $\{x^n\}$ converges to a point of P, x^*.

Furthermore,

$$\| x^n - x^* \| \leq 2d(x^0, P)\, \theta^n,$$

where $\Theta \in (0, 1)$ and depends only on the matrix A, where

$$A = \begin{pmatrix} a'_1 \\ a'_2 \\ \cdot \\ \cdot \\ \cdot \\ a'_m \end{pmatrix}$$

and $\qquad m = |I|.$

Agmon explicitly proved the convergence for the projection method. He also showed that the results could be extended to the case $\lambda \in (0, 2)$, where $\lambda^n = \lambda$ for all n.

Motzkin and Schoenberg, Eaves

Motzkin and Schoenberg [31] (among others) described a reflexion method for finding a point of P. They showed that if P has an interior then the sequence terminates with a point in P after a finite number of steps. Let $x^0 \notin P$ be an arbitrary starting point and generate a sequence as follows:

If $x^n \in P$ stop

If $x^n \notin P$ select a halfspace such that $x^n \notin H_j$

Let x^{n+1} be obtained by reflecting x^n in E_j. After finitely many iterations, x^ℓ will fall in P if P has an interior.

The general reflecion function can be expressed as follows:

For $i = 1, \ldots, m$ define $f_i: R^k \to R^k$ by

$$f_i(x) = x + 2d(x, H_i)a_i = \begin{cases} x - 2(a_i x + b_i)a_i & \text{if } x \notin H_i \\ x & \text{if } x \in H_i \end{cases}$$

Let $g_1, g_2, \ldots$ be a sequence where

$g_j \in \{f_1, \ldots, f_m\}$ for $j = 1, 2, \ldots$

Define g^n to be the composite function

$$g^n(x) = g_n(g_{n-1}\ (\ldots(g_1(x))\ \ldots))$$

(2.6) Theorem (Motzkin and Schoenberg)

Let $P \neq \emptyset$ and $I \neq \emptyset$ and finite. If $P = \bigcap_{i \in I} H_i$ has an interior, then for any x^0, there is an ℓ such that

$g^\ell(x) \in P.$

Eaves [11] extended this result and demonstrated that the reflection procedure has a uniform property. Namely, if x^0 is within

a distance δ of P, then $g^{\ell}(x^0)$ will fall in P for some ℓ where $\ell \leq \ell^*$ and ℓ^* depends on δ and not on x^0.

(2.8) Theorem (Eaves)

Assume that $P = \bigcap_{i \in I} H_i$ has an interior. Let X be the set of points within a distance δ of P, that is,

$$X = \{x \in R^k : d(x, P) \leq \delta\}$$

Then there is an ℓ such that $g^{\ell}(X) \subset P$. g^{ℓ} is thus a piecewise, linear retraction from X to P.

<u>Piecewise and linear</u> means that it has finitely many linear pieces. <u>Retraction</u> means $g^{\ell}: X \to P$ is continuous, $P \subset X$ and $g^{\ell}(x) = x$ for $x \in P$. See Figure 4 for example. Six points chosen arbitrarily within 1" of P all converge in under four iterations.

<u>Goffin</u>

<u>Definitions</u>. Goffin's [15] work deserves special mention. He has provided a comprehensive treatment of relaxation methods and presented several useful finiteness properties. We need a few more definitions before presenting Goffin's results.

A convex subset F of P is a <u>face</u> of P if

$$\left.\begin{array}{r} x, y \in P \\ (x, y) \quad F \neq \phi \end{array}\right\} \to x, y \in F.$$

We denote the set of faces of P by $\mathcal{F}(P)$. Figure 5 illustrates $\mathcal{F}(P)$ for a given P. Zero-dimensional faces are 1, 2, 3. One-dimensional faces are 4, 5, 6. The polytope 7 itself is also a face.

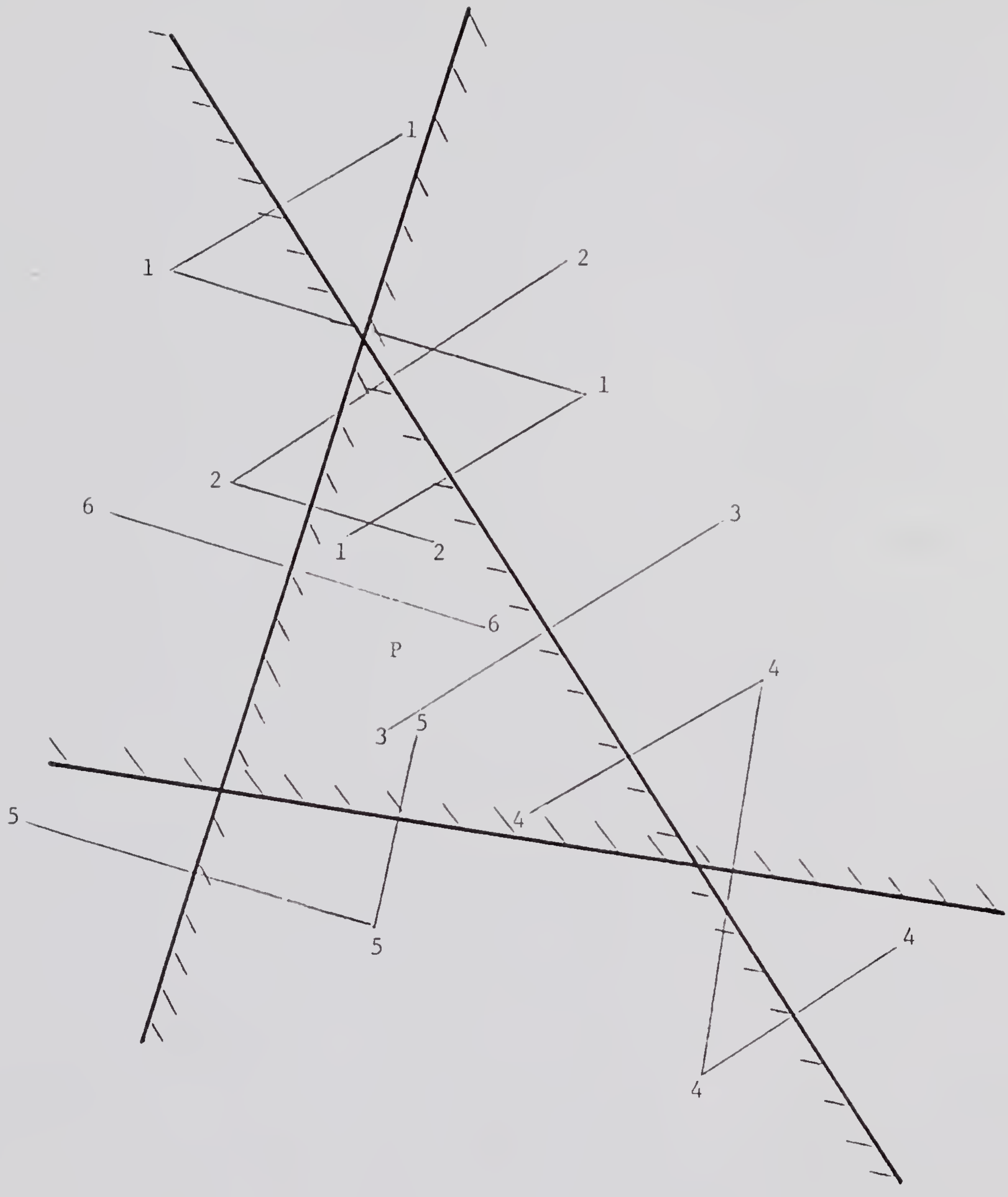


FIGURE 4

The zero-dimensional faces of P are called the extreme points of P. The n-1 dimensional faces are called facets of P.

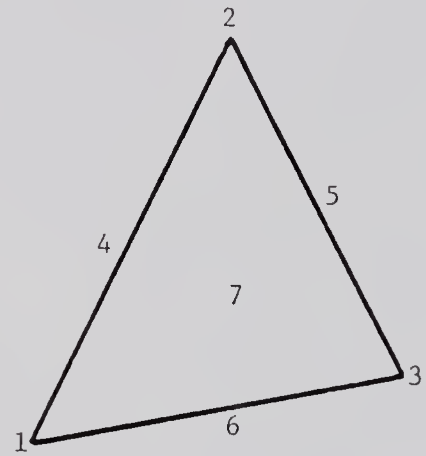


FIGURE 5

$N_S(x)$ defined by

$$N_S(x) = \{u \in R^k: u'(z - x) \leq 0, \forall z \in S\}$$

is the normal cone to S at x. $N_S(x)$ is a non-empty, closed convex cone. It is non-empty because it contains at least the origin. It is closed and convex because the intersection of closed halpspace is closed and convex. It is a cone because for all $u \in N_S(x)$, $\lambda u \in N_S(x)$, $\lambda \geqq 0$.

$$C_S(x) = (N_S(x))^p$$

is the supporting cone to S at x. Figure 6 illustrates these last two definitions.

A point x of a set P is a relative interior point designated r.i.(P.) if it is an interior point of P with respect to the relative

topology induced in $M(P)$ by the Euclidean topology, i.e., $x \in$ r.i.(P) if there exists $\delta > 0$ such that $B_\delta(x) \cap M(P) \subset P$. For example, as shown in Figure 7, a line segment in R^2 has no interior in R^2 but has a relative interior.

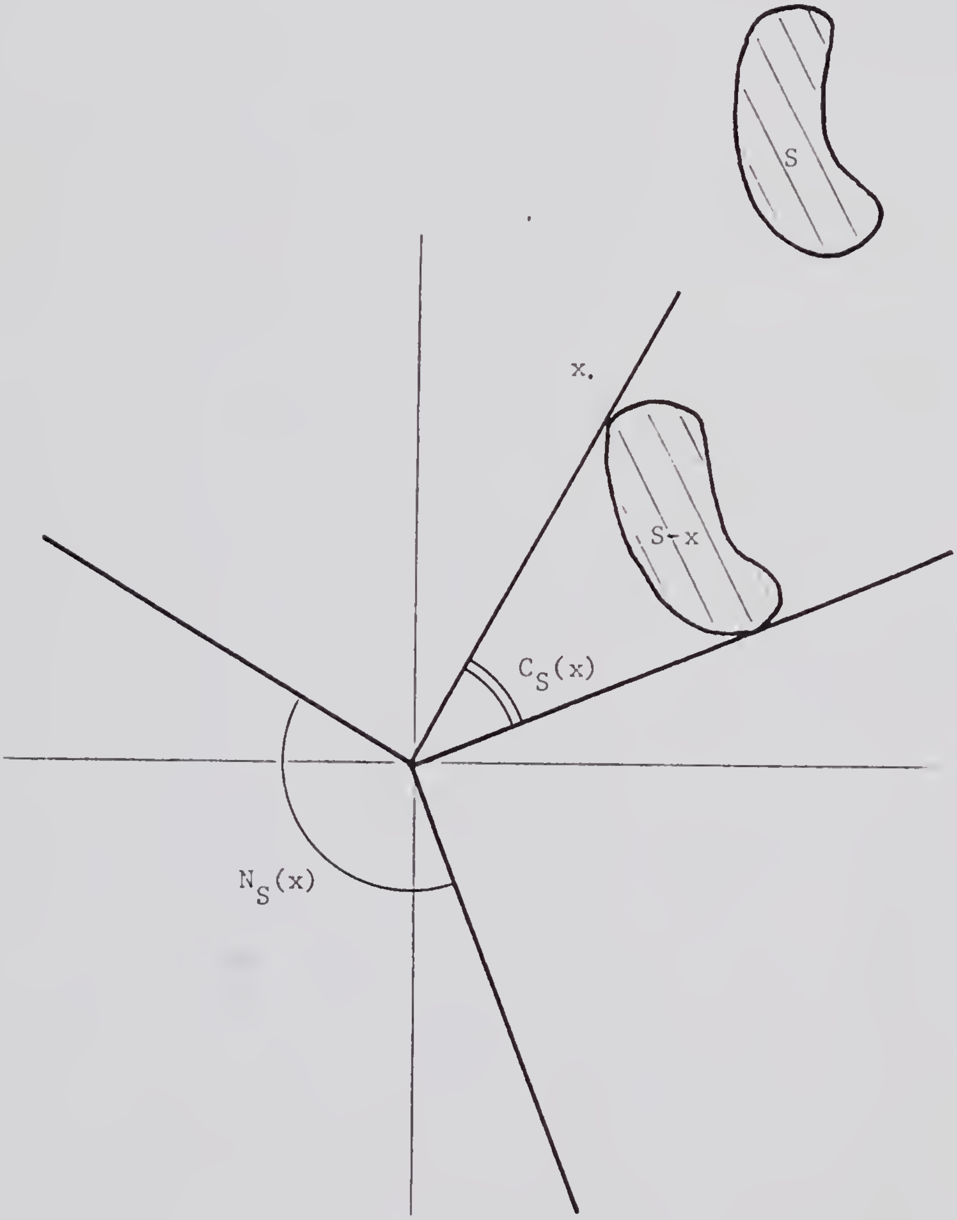


FIGURE 6

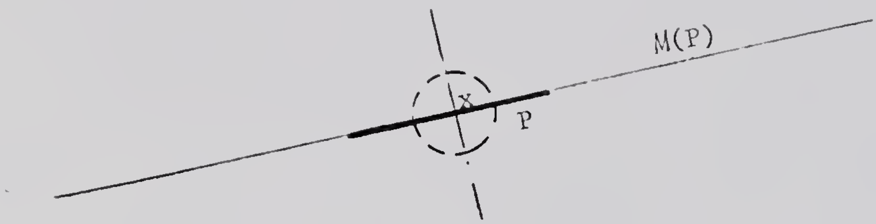


FIGURE 7

It can be shown that the normal cone is the same for all relatively interior points of a given face, i.e.,

$$N_P(x) = N_P(F) \text{ for any } x \in \text{r.i.}(F)$$

where F is a face of P. Similarly

$$C_P(x) = C_P(F) \text{ for any } x \in \text{r.i.}(F).$$

Let T be any set, then -T is defined as

$$-T = \{x \in R^k: x = -y,\ y \in T\}$$

A closed convex set is said to be <u>smooth enough at a point y</u> of its boundary if

$$-N_P(y) \subset C_P(y)$$

A closed convex set is said to be <u>smooth enough</u> if it is smooth enough at every boundary point, or equivalently if

$$-N_P(F) \subseteq C_P(F) \qquad \forall\, F \in \mathcal{F}(P) - \emptyset$$

where $\mathcal{F}(P)$ stands for the set of faces of P and $\mathcal{F}(P) - \phi$ is the set $\mathcal{F}(P)$ with ϕ removed.

The smooth enough property applied to a polyhedron would require all its "corners" to be "smooth enough," carrying the analogy to k-dimensional Euclidean space.

Some characteristics of smooth enough convex sets are mentioned below:

> A polyhedral cone C is smooth enough if and only if (iff) $-C^P \subset C$, where C^P is the polar of C. $C^P=\{u \in R^k: u'z \leqq 0, \forall z \in C\}$.
>
> A polytope (bounded polyhedron) is smooth enough iff all its extreme points are smooth enough.
>
> A polyhedron is smooth enough iff $-N_P(F) \subset C_P(F)$ for all the minimal elements of the poset $\{F(P) - \phi, \subset\}$ where $\subset$ stands for set inclusion.

Some illustrations are given in Figure 8.

A poset $(P, \leq)$ is a set P on which a binary relation $\leq$ is defined such that it satisfies the following relations:

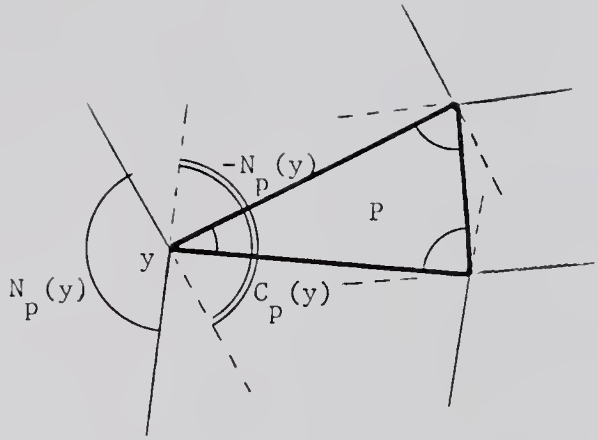
(i) for all x, $x \leq x$ (reflexivity)

(ii) if $x \leq y$ and $y \leq z$ then $x \leq z$ (transitivity)

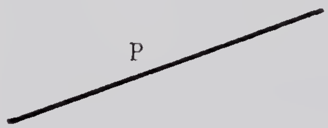
(iii) if $x \leq y$ and $y \leq x$ then $x = y$ (antisymmetry)

Examples of posets are $\{R, \leq\}$, i.e., the reals with the ordinary less than equal to operation, a dictionary with lexicographic ordering, and the power set S, P(S), with set inclusion, $\subset$, as the binary operation.

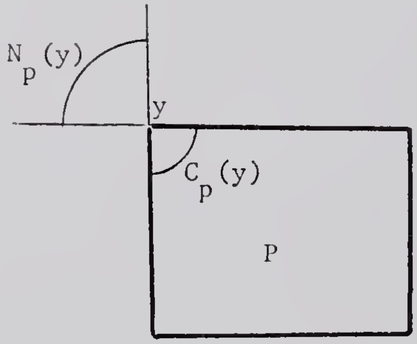
Given a unit vector $e \in R^k$ and an angle $\alpha \in [0,\Pi]$, the set of vectors which make an angle with e less than or equal to α is called the <u>spherical cone</u>, $C_\alpha(e)$, with axis e and half aperture angle α.



(a) Not smooth enough at the extreme points of P.



(b) Not smooth enough at any point of P.

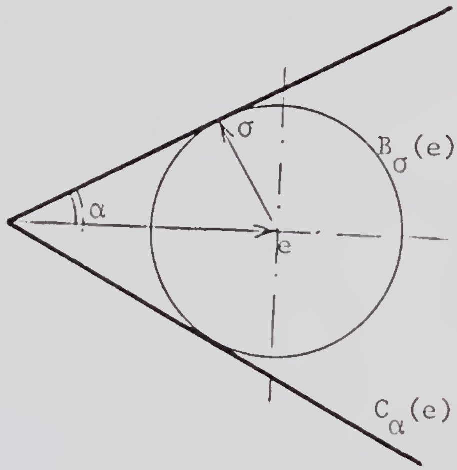


(c) Smooth enough.

FIGURE 8

$$C_\alpha(e) = \{x \in R^k : x'e \geqq \| x \| \text{Cos } \alpha\}$$

See Figure 9 for an illustration



$$C_\alpha(e) = c\{B_\sigma(e)\}$$

$$\sigma = \sin \alpha, \ \alpha \in [0, \tfrac{\pi}{2})$$

FIGURE 9

Some properties of $C_\alpha(e)$ are given below:

$C_\alpha(e)$ is closed for $\alpha \in [0,\Pi]$

$C_\alpha(e)$ is a convex set for $\alpha \in [0, \Pi/2]$

For $\alpha \in [0,\Pi/2)$ if we let $v = \text{Sin } \alpha$, then

$C_\alpha(e) = c\{B_v(e)\}$ where

$$c(S) = \{x \in R^k : x = \sum_{i \in L} \lambda^i x^i, \ x^i \in S, \ \lambda^i \geqq 0, \ L \text{ finite}\}$$

$c(S)$ is the convex cone hull of S.

Following Goffin [15] we define an obtuseness index $\upsilon(C)$ of a closed convex cone C to be the sine of the half aperture angle of the largest spherical cone contained in C. Properties of $\upsilon(C)$ are:

$$\upsilon(C) = \sup \{\text{Sin } \alpha : C_\alpha(e) \subset C,\ e \in B_1(0)\}$$

$$= \sup_{e\in S_1(0)} \min_{i\in I} (a_i' e)$$

$\upsilon(C) > 0$ iff C has an interior

$C_1 \subset C_2 \dashrightarrow \upsilon(C_1) \leqq \upsilon(C_2)$

$\upsilon(C) = 1$ iff C is a halfspace.

If $C_\alpha(e)$ is a spherical cone then $\upsilon(C_\alpha(e)) = \text{Sin } \alpha$.

The obtuseness index of a polyhedron P is defined by

$$\upsilon(P) = \inf_{x\in \partial P} \upsilon(C_P(x)) = \min_{F \in F(P)-\phi} \upsilon(C_P(F))$$

For a polytope we have

$$\upsilon(P) = \min_{F \in \text{Vertices of } P} \upsilon(C_P(F))$$

If $\upsilon(P) \geqq 1/\sqrt{2}$, then P is smooth enough. However, this is not a necessary condition for P to be smooth enough. If P is the positive orthant in R^3, then P is smooth enough but

$$\upsilon(P) = 1/\sqrt{3} < 1/\sqrt{2}.$$

The distance from a point x to a set $S \subset R^k$ is defined by

$$d(x, S) = \inf_{z\in S} ||x - z||$$

If S is closed, then the infimum is attained, and the set of points of S where it is attained is the set of points closest to x in S

$$T_S(x) = \{y \in S: ||x - y|| = d(x, S)\}$$

If a set K is closed and convex, then $T_K(x)$ reduces to a single point

which will also be denoted by $T_K(x)$. This point is called the <u>projection of x into K</u>. Hence $T_K(x)$ is a retraction from R^k to K.

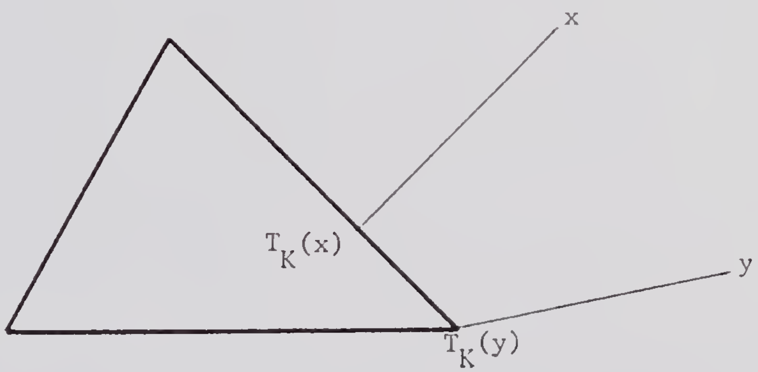
Let K be a closed convex set in R^k. The following important result implies the continuity of the map T_K.

(2.8) Theorem (Cheney, Goldstein)

The projection map T_K satisfies the Lipschitz condition (for C = 1). That is

$$\| T_K(x) - T_K(y) \| \leq \| x - y \|$$

See Figure 10 for an illustration:



$$\| T_K(x) - T_K(y) \| \leqq \| x - y \|$$

FIGURE 10

(2.9) Theorem (Goffin)

Let K be a closed convex set of R^k and $T_K(x)$ the unique closest point to x in K. Then

$$\{[T_K(x) + N_K(T_K(x))],\ T_K(x) \varepsilon K\} \text{ is a partition of } R^k.$$

See Figure 11 for an illustration.

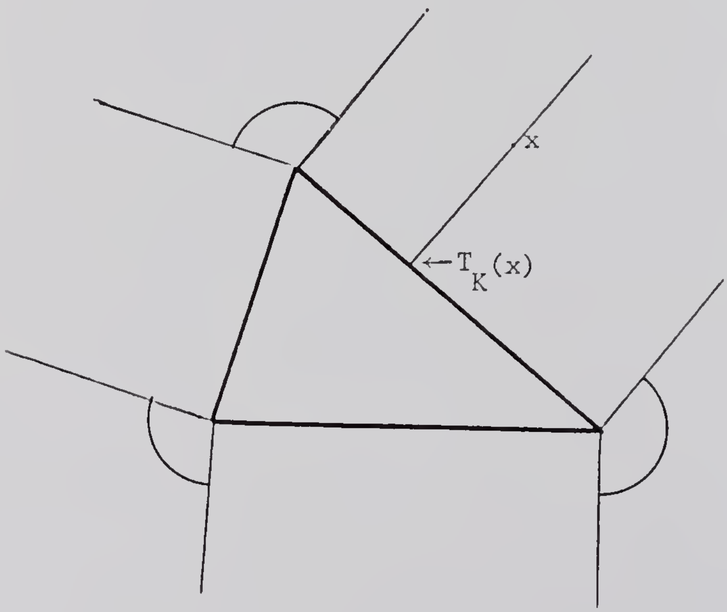


FIGURE 11

(2.10) Lemma

Let $C \subset R^k$ be a closed convex cone. Then

$$T_C(-C^P) \subset -C^P$$

where C^P is the polar of C.

An alternative formulation of a spherical surface with axis M, radius d(x, M) and center $T_M(X)$ (see 2.3) is contained in the following lemma.

(2.11) Lemma (Goffin)

Let M be affine in R^k and $x \in R^k$, then

$$S_{d(x,M)}(M, T_M(x)) = \{y \in R^k;\ \|x - z\| = \|y - z\|,\ \forall z \in M\}$$

Finally a well known result on norm-equivalence is stated below. Here we state the result in terms of distances from x to P with $x \notin P$.

(2.12) Lemma

Let $P = \bigcap_{i \in I} H_i$, $P \neq \emptyset$, $I \neq \emptyset$, then there exists a scalar $0 < \mu \leq 1$ such that for all $x \in R^k$

$$\mu d(x, P) \leq d_p(x) \leq d(x, P).$$

This result is illustrated in Figure 12.

Another Proof of Agmon's Result. The following is a different proof for the result originally due to Agmon. We will find this useful in subsequent parts of this material. In this proof we make use of the result that the sequence generated by the relaxation method is Fejér-monotone with respect to P.

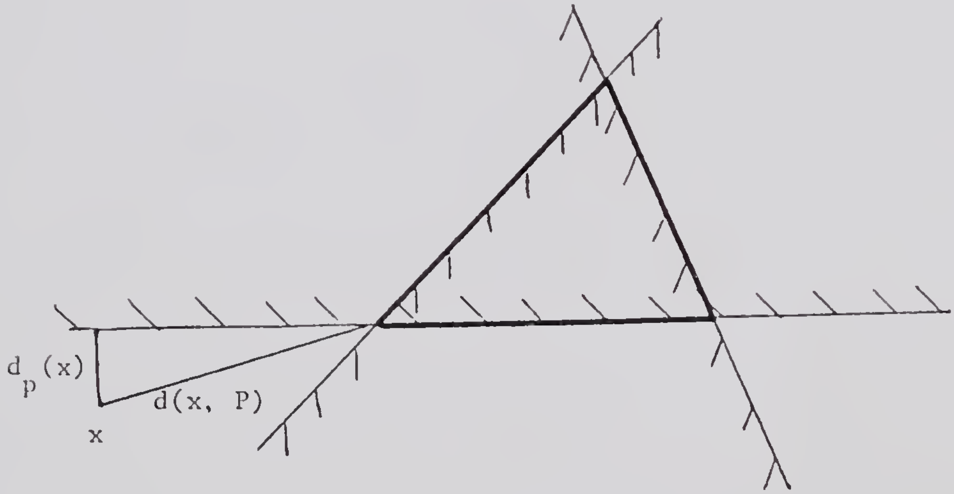


FIGURE 12

(2.13) Theorem

Let $P = \bigcap_{i \varepsilon I} H_i$, $P \neq \emptyset$, $I \neq \emptyset$ and finite, then the relaxation method generates a sequence which converges finitely or infinitely to a point $x^* \varepsilon P$ for any given value of the relaxation parameter λ such that $\lambda \varepsilon (0, 2)$.

Furthermore, we have

$$\| x^n - x^* \| \leqq 2d(x^0, P)\, \theta^n$$

where $\quad \theta = [1 - \lambda(2 - \lambda)\mu^2]^{1/2}$

$$\lambda \varepsilon [0,1)$$

Proof:

Assume $x^n \notin P$

First we will show using Figure 13 that

$$d^2(x^{n+1}, P) = d^2(x^n, P) - \lambda(2 - \lambda) d_P^2(x^n)$$

where $d_P(x^n)$ is the maximal distance from x^n to P i.e., $d_P(x^n) = \max_{i \varepsilon I} d(x, H_i)$.

From the right triangle x^{n+1}, t^n, $T_P(x^{n+1})$ we have:

$$d^2(x^{n+1}, P) = d^2(x^{n+1}, t^n) + d^2(t^n, P) \text{ ....(1)}$$

From the right triangle x^n, t^n, $T_P(x^n)$ we have:

$$d^2(x^{n+1}, P) = d^2(x^n, t^n) + d^2(t^n, P) \text{ ....(2)}$$

From (1) and (2)

$$d^2(x^{n+1}, P) - d^2(x^n, P) = d^2(x^{n+1}, t^n) - d^2(x^n, t^n)$$

$$\begin{aligned} d^2(x^{n+1}, P) &= d^2(x^n, P) - [d(x^{n+1}, t^n) + d(x^n, t^n)] \\ &\quad [d(x^n, t^n) - d(x^{n+1}, t^n)] \\ &= d^2(x^n, P) - \lambda d_P(x^n)[(2 - \lambda)d_P(x^n)] \\ &= d^2(x^n, P) - \lambda(2 - \lambda)d_P^2(x^n) \end{aligned}$$

It will be noticed that our proof is independent of Figure 13. The figure only helps to illustrate the ideas.

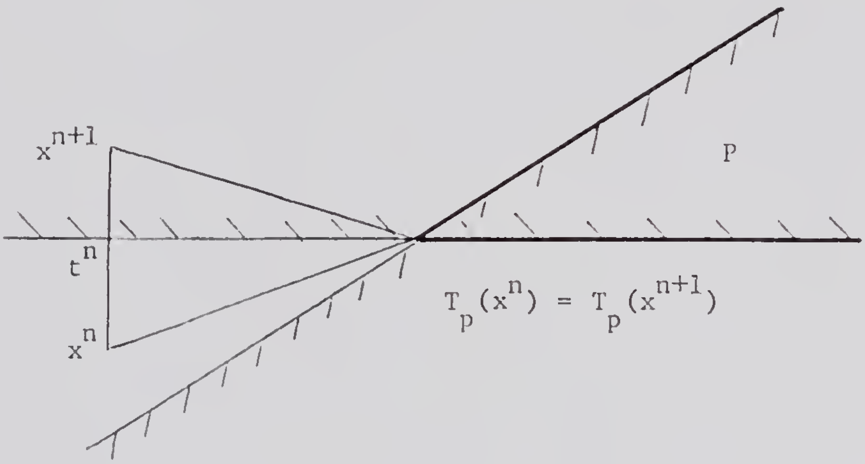


FIGURE 13

By lemma 2.12 there exists a $\mu > 0$ such that

$$\mu d(x, P) \leqq d_p(x) \leqq d(x, P)$$

hence

$$d^2(x^{n+1}, P) \leqq d^2(x^n, P) - \mu^2\lambda(2 - \lambda)d^2(x^n, P)$$
$$= d^2(x^n, P)[1 - \lambda(2 - \lambda)\mu^2]$$

(2.14) Let $\theta = [1 - \lambda(2 - \lambda)\mu^2]^{1/2}$

$\theta = 0$ when $\lambda = 1$, $\mu = 1$. In general $\theta \in [0, 1)$. Thus,

$$d^2(x^{n+1}, P) \leqq \theta^2 d^2(x^n, P)$$

or $$d(x^n, P) \leqq \theta^n d(x^0, P)$$

Case I: If the sequence does not terminate

$$\lim_{n\to\infty} d(x^n, P) = 0$$

Since $\{x^n\}$ is Fejér-monotone and dim P = n, then $\{x^n\}$ converges to $x^* \in \partial P$ by Theorem 2.4 where ∂P is the boundary of P.

Case II: If the sequence terminates in a finite number of steps $\|x^{n+1} - z\| \leqq \|x^n - z\|$, $\forall z \in P$ since the sequence is Fejér-monotone with respect to P. Clearly $\|x^* - T_P(x^n)\| \leqq \|x^n - T_P(x)\|$, hence

$$x^* \in B_{d(x^n,\ P)} T_P(x^n)$$

and x^*, x^n both belong to the ball $B_{d(x^n,\ P)} T_P(x^n)$ (see Figure 14). Hence,

$$\begin{aligned} \|x^n - x^*\| &\leqq 2d(x^n, P) \\ &\leqq 2\theta^n d(x^0, P) \end{aligned}$$

If the sequence terminates, we can take x^* to be the last point of the sequence. ///

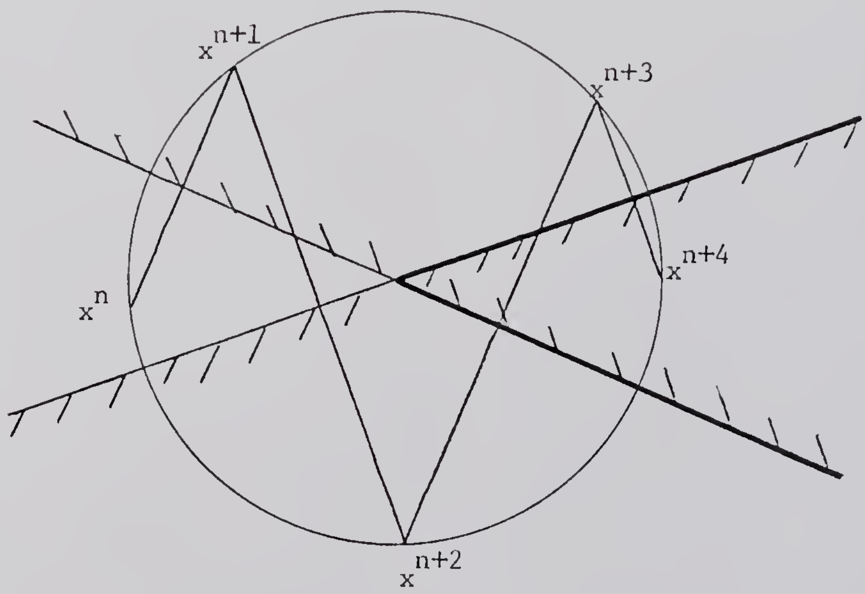


FIGURE 14

In the above procedure, if $1 \leq \lambda < 2$ then

$$P^n = \{x: a'_{i_n} x + b_{i_n} \geq 0\}$$

That is, P^n is the closed halfspace defined by the constaint i_n.
If $0 < \lambda < 1$, then

$$P^n = \{x: a'_{i_n} x + (1 - \lambda)b_{i_n} - \lambda a'_{i_n} x^n \geq 0\}$$

and again $P \subset P^n$. In both cases P^n is a relaxed form of P and at iteration n we find $x^n \varepsilon P^n$.

Range of λ for finite convergence. Using relation 2.14 giving θ against different values of λ and μ, we get the results in Table 1 shown graphically in Figure 15.

TABLE 1

$\mu = 1$	λ	.1	.5	.75	1	1.5	1.9
	$\theta_1 = f(\lambda)$	.9	.5	.24	0	.5	.9
$\mu = \frac{1}{2}$	λ	.1	.5	.75	1	1.5	1.9
	$\theta_{\frac{1}{2}} = f(\lambda)$	.98	.9	.87	.87	.9	.98
$\mu = \frac{1}{4}$	λ	.1	.5	.75	1	1.5	1.9
	$\theta_{\frac{1}{4}} = f(\lambda)$	.99	.98	.97	.97	.98	.99

From Figure 15 it would appear that $\lambda = 1$ gives best results (lowest θ). However, Motzkin and Schoenberg demonstrated that when P has an interior $\lambda = 2$ leads to finite convergence. Also if P has acute angles, μ will be low and θ high. Thus an obtuse angle should lead to faster convergence. Goffin determined the range of values of λ which gives finite convergence.

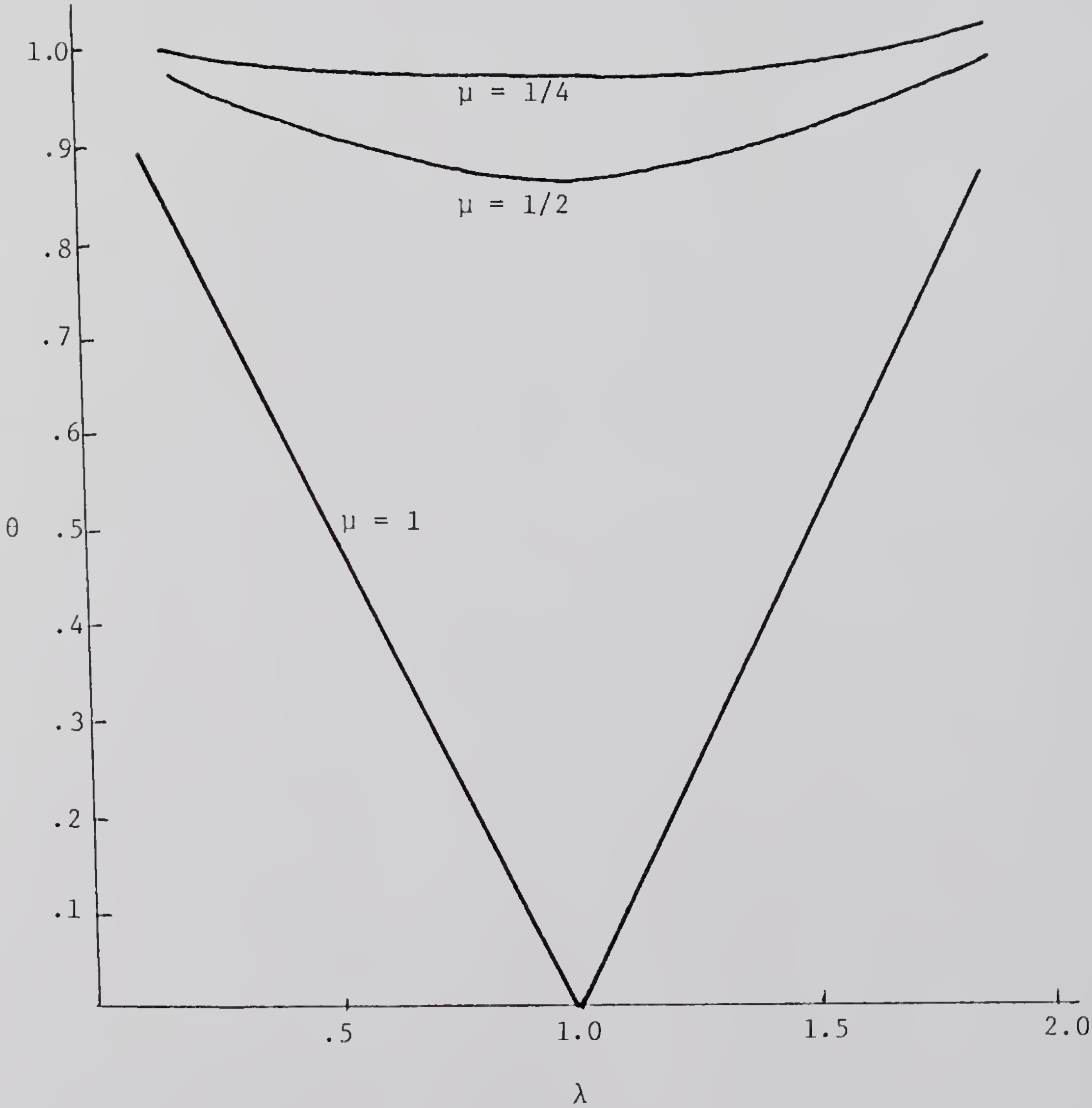


FIGURE 15

(2.15) Theorem (Goffin)

Let P be a polyhedron with an interior. Then the sequence generated by the relaxation method converges finitely to a point of P if

(a) P is smooth enough and $\lambda \in [1, 2]$

(b) $\lambda \in (\lambda^*, 2]$

where

$$\lambda^* = \frac{2}{1 + \upsilon(P)} \text{ and } \upsilon(P) \text{ is the obtuseness index of P.}$$

Furthermore, if $\lambda > 2$ then the sequence either converges finitely to a point of P or it does not converge.

The first part of the theorem shows that if all the corners of P are smooth enough then the range of λ over which we can get convergence in a finite number of steps is [1, 2].

The second part of the theorem shows that for a polyhedron that is not smooth enough, the extent by which we can deviate from $\lambda = 2$ depends on the acuteness of the most restrictive extreme point of P -- the more acute it is, the less we can relax λ from a value of 2.

<u>Merzlyakov</u>

<u>Definitions and Motivation</u>. The final work on relaxation methods that we review is by Merzlyakov [30]. Merzlyakov's method takes advantage of the fact that convergence properties of relaxation methods improve if, instead of considering only the most violated constraint, as was done by Agmon, among others, a larger number of supporting planes are considered. We would like to emphasize this unusual property of

relaxation methods -- redundancy, in terms of number of constraints, improves the convergence rate. Additional halfspaces have the effect of increasing μ which lowers θ, the convergence ratio.

At each $x \in R^k$ let

$$V(x) = \{i: a_i'x + b_i < 0, i = 1, \ldots, m\}$$

That is, $V(x)$ is the set of indices of constraints violated by x. A cavity C_V is the set

$C_V = \{x \in R^k: V(x) = V\}$ where V is a subset of first m natural numbers. A subcavity S_V^i of C_V is the subset of all $x \in C_V$ which violate constraint $i \in V$ no less than any other constraint $j \in V$. Ties are broken by the natural ordering. Notice that subcavities along with P finitely partition R^k. Figure 16 illustrates the above.

Associate with each subcavity S_V^j a fixed $\lambda_i(j, V)$ where

$$\lambda_i(j, V) = \begin{cases} 0 \text{ if } i \notin V \\ \geqq 0 \text{ if } i \in V \\ > 0 \text{ if } i = j \end{cases}.$$

For example for subcavity $S_{\{2,3\}}^2$ (see Figure 16).

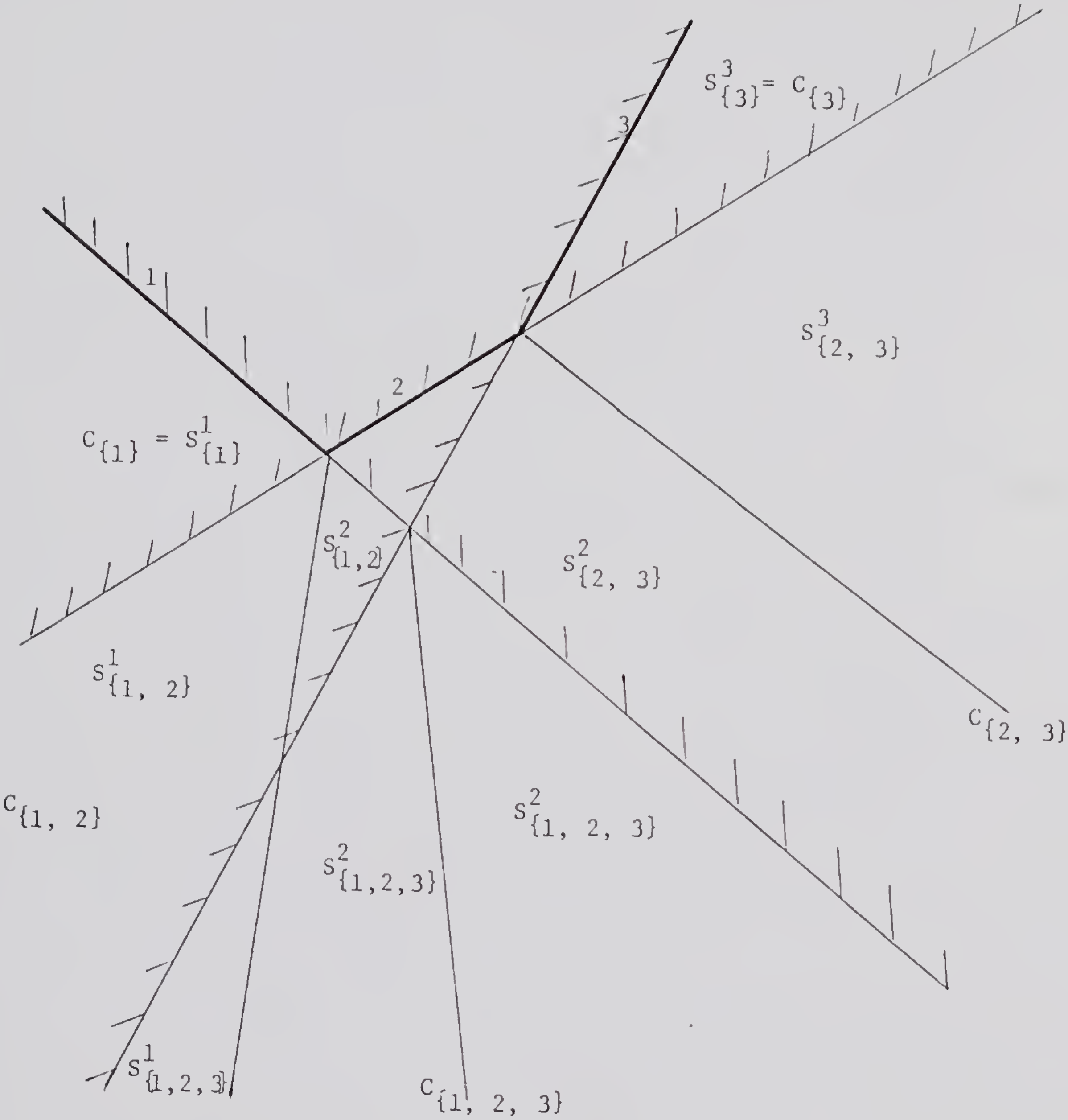
$$\lambda_1(2, \{2, 3\}) = 0$$

$$\lambda_2(2, \{2, 3\}) > 0$$

$$\lambda_3(2, \{2, 3\}) \geqq 0$$

<u>Method</u>. Merzlyakov's relaxation procedure is as follows:

Let $\lambda \in (0, 2)$ and x^0 specified. If at iteration n, $x^n \in P$ stop. Otherwise, let



$S^3_{\{3\}} = C_{\{3\}}$
3
1
2
$S^3_{\{2, 3\}}$
$C_{\{1\}} = S^1_{\{1\}}$
$S^2_{\{1,2\}}$
$S^2_{\{2, 3\}}$
$S^1_{\{1, 2\}}$
$C_{\{2, 3\}}$
$C_{\{1, 2\}}$
$S^2_{\{1, 2, 3\}}$
$S^2_{\{1,2,3\}}$
$S^1_{\{1,2,3\}}$
$C_{\{1, 2, 3\}}$

FIGURE 16

$$(2.16)\qquad x^{n+1} = x^n - \frac{\lambda[\Sigma\lambda_i(j,V)(a_i'x^n+b_i)][\Sigma\lambda_i(j,V)a_i]}{[\Sigma\lambda_i(j,V)a_i]'[\Sigma\lambda_i(j,V)a_i]}$$

where $x^n \varepsilon S_V^j$.

We again make the assumptions that $P \neq \emptyset$ and without loss of generality assume $||a_i|| = 1$ for each $i = 1, \ldots, m$. Also assume without loss of generality that $\Sigma\lambda_i(j,V) = 1$. Goffin showed that if $P \neq \emptyset$ then the vector $\Sigma\lambda_i(j,V)a_i$ cannot be zero.

Merzlyakov showed that the procedure given above converges to a point of P with a linear rate of convergence. If $1 \leqq \lambda < 2$ then we can say

$$P^n = \{x: \sum_i \lambda_i(j,V)(a_i'x + b_i) \geqq 0\}$$

and clearly $P \subset P^n$. If $0 < \lambda < 1$, then

$$P^n = \{x: \sum_i \lambda_i(j,V)[a_i'x + (1 - \lambda)b_i - \lambda(a_i'x^n)] \geqq 0\}$$

and $P \subset P^n$. Here again P^n is a relaxed form of P and the sequence obeys our stipulation for relaxation method, i.e., at iteration n we find $x^n \varepsilon P^n$ where $P \subset P^n$.

The relaxation sequence $\{x^n\}$ congerges infinitely to a point $x^* \varepsilon \partial P$, where ∂P is the boundary of P, only if the whole sequence $\{x^n\}_{n \geqq N}$ "jams" into $x^* + N_P(x^*)$. By moving in a direction that is a convex combination of violated constraints we force the sequence out of the jam. For this reason Merzlyakov method can be referred to as an antijamming procedure. Table 2 and Figure 17 illustrate this point. The table and figure are based on the example given in Merzlyakov's paper.

TABLE 2

Consider the following system of linear inequalities

$$-x_1 + 10x_2 + 10 \geq 0$$

$$3x_1 - 10x_2 - 30 \geq 0$$

Let $x^0 = (0, 0)$. We solve this by

(a) Agmon's method (b) Merzlyakov's method

where $\lambda = 3/4$, $\| a_{i_n} \| = 1$

(a) Agmon's method

$$x^{n+1} = x^n - \lambda(a'_{i_n} x^n + b_{i_n}) a_{i_n} ;$$

n	x^n: x_1^n	x^n: x_2^n	$a'_{i_n} x^n + b_{i_n}$	a_{i_n}	
0	.000	.000	-2.873	$\binom{.287}{-.958}$	
1	.619	-2.064	-1.121	$\binom{-.099}{.995}$	
2	.536	-1.228	-1.543	$\binom{.287}{-.958}$	
3	.868	-2.337	-1.416	$\binom{-.099}{.995}$	
4	.763	-1.280	-1.429	$\binom{.287}{-.958}$	
5	1.071	-2.306	-1.406	$\binom{-.099}{.995}$	
6	.966	-1.257	-1.392	$\binom{.287}{-.958}$	
7	1.266	-2.257	-1.377	$\binom{-.099}{.995}$	
8	1.163	-1.230	-1.362	$\binom{.287}{-.958}$	
9	1.456	-2.208			

TABLE 2 (continued)

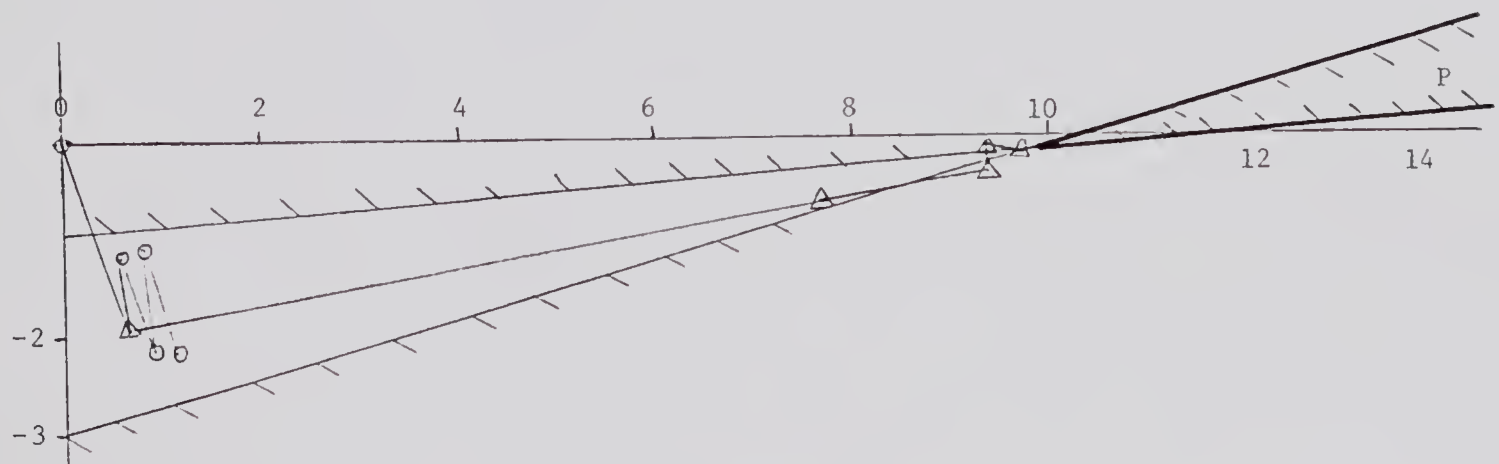
(b) Merzlyakov Method

$$x^{n+1} = x^n - \lambda \frac{[\Sigma\lambda_i(j,V)(a_i'x^n + b_i)][\Sigma\lambda_i(j,V)a_i]}{[\Sigma\lambda_i(j,V)a_i]'[\Sigma\lambda_i(j,V)a_i]}$$

n	x^n		$\Sigma\lambda_i(j,V)(a_i'x^n + b_i)$	$\Sigma\lambda_i(j,V)a_i$	$[\Sigma\lambda_i(j,V)a_i]'[\Sigma\lambda_i(j,V)a_i]$
	x_1^n	x_2^n			
0	.000	.000	-2.873	$\binom{.287}{-.958}$	1
1	.619	-2.064	-1.838	$\binom{.188}{.037}$	.037
2	7.622	-.686	-.459	$\binom{.188}{.037}$	.037
3	9.371	-.342	-.273	$\binom{-.099}{.995}$	1
4	9.351	-.138	-.125	$\binom{.188}{.037}$	.037
5	9.827	-.044	-.032	$\binom{.188}{.037}$	.037
6	9.949	-.020	-.010	$\binom{-.099}{.955}$	1
7	9.948	-.013	-.008	$\binom{.188}{.037}$	.037
8	9.978	-.007	-.002	$\binom{.287}{-.958}$	1
9	9.978	-.008			

Notice that the Merzlyakov method is much more effective than the Agmon procedure. Convergence to a point of P appears much faster with the Merzlyakov Method.

Generalized Merzlyakov Procedure. The Merzlyakov procedure can be generalized [24] by allowing the λ_i's to be chosen based on the current point rather than fixed for a subcavity. For any x where $V(x) \neq \emptyset$ let



△ moves by Merzlyakov

○ moves by Agmon

$H_1 = \{\binom{x_1}{x_2} : -x_1 + 10x_2 + 10 \geq 0\}$

$H_2 = \{\binom{x_1}{x_2} : 3x_1 - 10x_2 - 30 \geq 0\}$

$P = H_1 \cap H_2.$

FIGURE 17

$$\lambda_i(x) = \begin{cases} 0 \text{ if } a_i'x + b_i > 0 \\ \geq 0 \text{ if } a_i'x + b_i \leq 0 \end{cases}$$

where $0 < \sum_i \lambda_i(x) \leq \beta$, $\lambda_i(x) > 0$ for some $i \in V(x)$ and at each n, there is a $j \in V(x^n)$ such that

$$\frac{\lambda_j(x^n)(a_i'x^n + b_j)}{a_{j*}'x^n + b_{j*}} \geqq \theta$$

where j* indexes a most isolated constraint and $\theta > 0$.

The convergence properties of this method are described in the next chapter.

Typical Subgradient Method

Procedure

Let f be a convex but not necessarily differentiable function defined on R^n. $u \in R^n$ is said to be a subgradient of f at x if it satisfies the following inequality:

$$f(y) \geqq f(x) + u'(y - x) \text{ for all } y \in R^n.$$

The set of subgradients of f at x is called the subdifferential of f at x and denoted by $\partial f(x)$.

$$\partial f(x) = \{u \in R : u \text{ is a subgradient of f at } x\}$$

x^* is a minimum of f if and only if $0 \in \partial f(x^*)$.

A typical subgradient algorithm works as follows:

(1) choose $x^0 \in R^n$ arbitrarily

(2) compute a subgradient u^n of f at x^n, i.e., $u^n \in \partial f(x^n)$.

If $u^n = 0$ an optimal point has been found. If not go to (3).

(3) Find x^{n+1} as follows: $x^{n+1} = x^n - t^n u^n$

where t^n is a scalar generally obtained by the approximate minimization of

$$f(x^n - tu^n) \text{ for } t \geqq 0.$$

Go to (2) with x^n replacing x^{n+1}.

The function f could be an arbitrary convex function. If the objective is to find a point of a polyhedron, one could choose a number of different functions. One choice could be

$$f(x^n) = \underset{i=1,...,m}{\text{Max}} (-a_i'x^n - b_i)$$

Since f is the maximum of a finite number of convex functions, f is convex. Oettli [32] has given another interesting way to define a function which can be used to find a point of a polyhedron. This is given latter.

Variations

Reverting back to the general subgradient procedure given by the iterative scheme

$$x^{n+1} = x^n - t^n u^n$$

various proposals [16] have been made for the step size t^n. The earliest one was by Polyak [34] who suggested $\Sigma t^n = \infty$ and $\lim_{n\to\infty} t^n = 0$. The merit of this procedure is that we could choose any arbitrary subgradient $u^n \in \partial f(x^n)$ and the sequence would converge to some x^* with the stipulated stepsize (e.g., with $t^n = \frac{1}{n}$). However convergence is dismally slow and the method is of little practical value. Polyak attributes the original idea of this method to Shor [36].

Another idea also due to Polyak [35] is to use

(2.17) $$t^n = \frac{\lambda_n [f(x^n) - \bar{f}]}{\|u^n\|^2}$$

with $\bar{f} = f(x^*)$ the optimum value of the function. Polyak has shown that with $0 < b_1 \leqq \lambda_n \leqq b_2 < 2$ the sequence converges to x^* with a linear rate of convergence. A variation of 2.17, also due to Polyak is to use $\bar{f} > f(x^*)$, i.e., an overestimate of the optimum $f(x^*)$. Here again if $0 < b_1, \leqq \lambda_n \leqq b_2 < 2$ the sequence converges at a linear rate but to the level set

$$S = \{x\colon f(x) \leqq \bar{f}\}.$$

Thus for the result to be of value we must choose close enough overestimates of $f(x^*)$ and this is difficult in practice.

A final variation on the theme of 2.17 is to use an underestimate $\bar{f} < f(x^*)$. Eremin [12] has shown that the sequence converges to x^* if $\lim_{n\to\infty} \lambda_n = 0$ and $\Sigma\lambda_n = \infty$. Here again convergence to the optimal solution is very slow.

Some Subgradient Procedures

Polyak

(2.18) Theorem (Polyak)

Let f be a quasi convex function to be minimized over R^k. If the sequence $\{x^n\}$ is obtained as follows

$$x^{n+1} = x^n - \lambda_n \frac{u^n}{\|u^n\|} \text{ with } \lambda_n = \|u^n\|, \text{ then there exists a}$$

subsequence $\{x^{n_k}\}$ such that $\lim_{k\to\infty} f(x^{n_k}) = f^*$

Proof: Select $\alpha > f^*$ and define

$$S_\alpha = \{x \in R^k\colon f(x) \leqq \alpha\}$$

S_α is convex (it is a level set). Hence there exists an $\tilde{x}$ in int S_α since $\alpha > f^*$ (see Figure 18). Choose $\rho > 0$ such that the neighborhood $B_\rho(\tilde{x}) \subset S_\alpha$.

Suppose $x^n \notin S_\alpha$ for all n

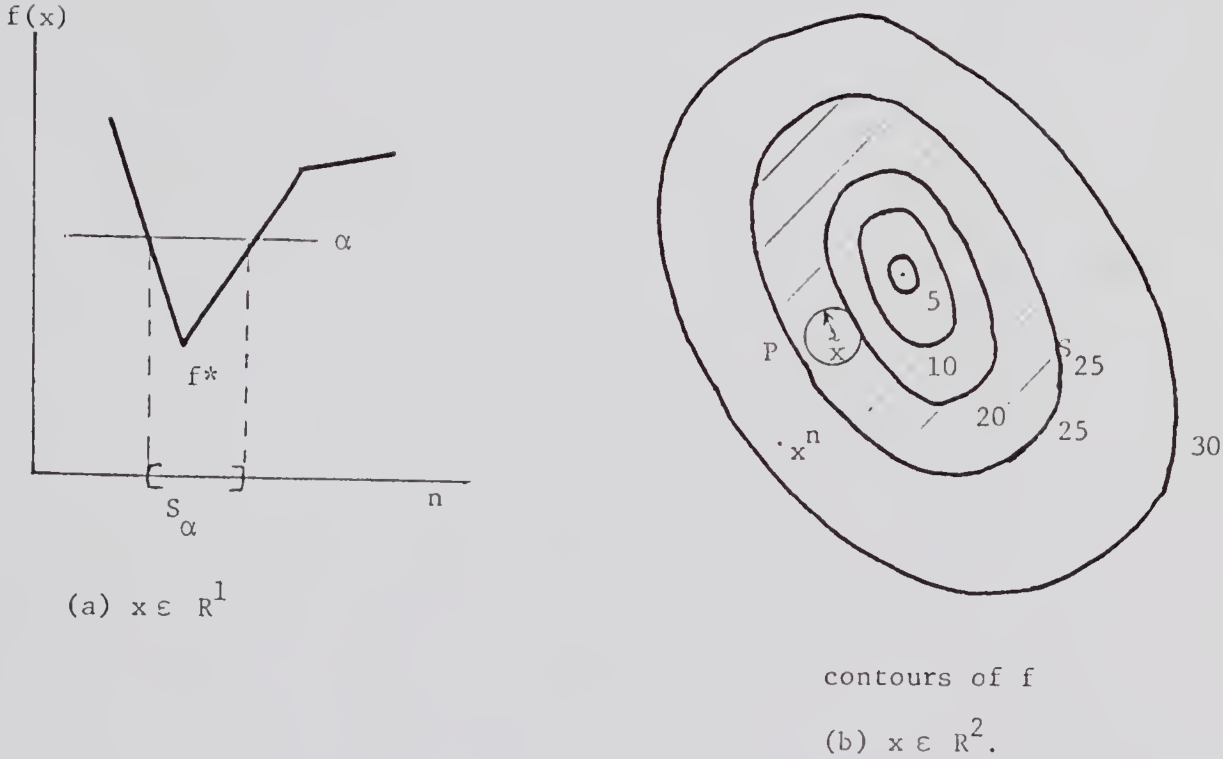


FIGURE 18

$$f(x) - f(x^n) \geqq u^{n'}(x - x^n), \ \forall x$$

since u^n is a subgradient at x^n. Thus

$$f(x^n) - f(x) \leqq u^{n'}(x^n - x).$$ Also

$$f(x) \leqq f(x^n) \quad \forall x \in S_\alpha$$

Combining the above

$$u^{n'}(x^n - x) \geqq f(x^n) - f(x) \geqq 0, \forall x \in S_\alpha$$

This holds for a point $\tilde{x} + \dfrac{\rho u^n}{\|u^n\|} \in B_\rho(\tilde{x}) \subset S_\alpha$.

Hence $\qquad u^{n'}x^n \geqq u^{n'}(\tilde{x} + \dfrac{\rho u^n}{\|u^n\|})$

or $\qquad \rho\|u^n\| \leqq u^{n'}(x^n - \tilde{x})$

But $\qquad \|x^{n+1} - \tilde{x}\| = \|x^n - u^n - \tilde{x}\|^2$

$$\begin{aligned}
&= \|x^n - \tilde{x}\|^2 + \|u^n\|^2 - 2u^{n'}(x^n - \tilde{x}) \\
&\leqq \|x^n - \tilde{x}\|^2 + \|u^n\|^2 - 2\rho\|u^n\| \\
&= \|x^n - \tilde{x}\|^2 + \|u^n\|^2 - 2\rho\lambda_n .
\end{aligned}$$

Choose N such that $\lambda_n \leqq \rho$, $\forall n \geqq N$. This is possible since $\lim_{n\to\infty} \lambda_n = 0$.

For $n = N$ to $n = N + m$

$$0 \leqq \|x^{N+m-1} - \tilde{x}\|^2 \leqq \|x^N - \tilde{x}\|^2 + \sum_{n=N}^{N+m} \lambda_n(\lambda_n - 2\rho)$$

$$\leqq \|x^N - \tilde{x}\|^2 - \rho \sum_{n=n}^{N+m} \lambda_n$$

For m large enough the right hand side is negative, since $\Sigma\lambda_n$ diverges -- a contradiction. Hence $x^n \in S_\alpha$.

Let $\lim_{k\to\infty} \alpha_k = f^*$. Then we have proved there exists $x^{n_k} \in S_{\alpha_k}$ such that

$$\lim_{k\to\infty} f(x^{n_k}) = f^*. \qquad ///$$

Computational experience with heuristic subgradient methods has been reported by Held and Karp [18], and Held, Wolfe, and Crowder [19].

Held, Wolfe, and Crowder use underestimates to $f(x^*)$. However the stepsize $\lambda^n = \lambda$ is kept constant for 2m iterations where m is a measure of the problem size. Then both λ and number of iterations are halved successively till the number of iterations reaches some value k, λ is then halved every k iteration. This sequence violates the requirement $\Sigma\lambda_n = \infty$ and it is possible that the sequence may converge to a wrong point. However in their computational experience this did not happen. Held, Wolfe and Crowder experimentally found that their procedure was effective for approximating the maximum of a set of piecewise linear concave functions. In their problems (assignment, multicommodity, and maxflow) the number of constraints is enormous, of the order of 100!. In all these cases they were able to get linear convergence (which is assured by the choice of stepsize).

Held and Karp used overestimates to $f(x^*)$ to obtain bounds for a branch and bound procedure for solving the traveling salesman problem. Using this procedure they were able to produce optimal solutions to all traveling salesman problems ranging in size upto 64 cities.

Oettli

Oettli proposed a subgradient method for finding a point of an arbitrary non-empty polyhedron. We will discuss his less general method which can be used to solve a linear program. As before, using the dual and an additional constraint expressing strong duality, a linear program can be stated to be the problem of finding an $x \in R^k$ satisfying

$$\ell_i^+(x) \geqq 0 \qquad i = 1, \ldots, m$$

where $\ell_i^+(x)$ represents the magnitude by which constaint i is violated at x; i.e.,

$$\ell_i^+(x) = \max\{0, -a_i'x - b_i\}$$

Define $\ell^+(x)$ to be the vector formed from violations of different constraints

$$\ell^+(x) = (\ell_1^+(x), \ell_2^+(x), \ldots, \ell_m^+(x))$$

Let $p(\cdot)$ be an isotone norm on R^m, i.e., if $0 \leqq x \leqq y$ then $p(x) \leqq p(y)$. It may be noted that all the usual norms like Euclidean and Tchebycheff are isotone norms. Define

$$d(x) = p(\ell^+(x))$$

Then clearly x satisfies $\ell_i^+(x) \geqq 0$ iff $d(x) = 0$.

Oettli showed that $d(\cdot)$ is a convex function and then described an iterative scheme using subgradients of $d(\cdot)$ that minimizes $d(\cdot)$ over R^k. This scheme is quite general since we have a different function for different norms.

(2.18) Lemma [Oettli]

$d(\cdot)$ is a convex function.

Proof:

$$\ell_j^+(z) \text{ is convex for all } j.$$

Thus $\ell^+(\alpha z^1 + \beta z^2) \leqq \alpha\ell^+(z^1) + \beta\ell^+(z^2)$ for $\alpha,\beta \geqq 0$, $\alpha+\beta = 1$.

Since p is monotonic

$$p(\ell^+(\alpha z^1 + \beta z^2) \leq p[\alpha\ell^+(z^1) + \beta\ell^+(z^2)]$$

and since p as a norm is convex

$$\leq p[\alpha\ell^+(z^1) + p[\beta\ell^+(z^2)]$$

$$= \alpha p[\ell^+(z^1)] + \beta\, p[\ell^+(z^2)]$$

hence $d(\alpha z^1 + \beta z^2) \leq \alpha d(z^1) + \beta d(z^2)$. ///

Let x^0 be given initially. If $x^n \varepsilon$ P stop. Otherwise, let

$$x^{n+1} = x^n - \frac{d(x^n)}{t(x^n)' t(x^n)} t(x^n) \tag{2.19}$$

where $t(x^n)$ is any subgradient of $d(\cdot)$ formed according to Oettli's rule at x^n. The sequence $\{x^n\}$ converges to a point of P with a linear rate of convergence.

(2.20) Lemma [Oettli]

Let $\gamma \varepsilon$ P, then for the Oettli's procedure

$$\| x^{n+1} - \gamma \|^2 \leqq \| x^n - \gamma \|^2 - \frac{d^2(x^n)}{\| t(x^n) \|^2}$$

Proof:

$$\| x^{n+1} - \gamma \|^2 = \| x^n - \frac{d(x^n)}{t(x^n)' t(n^n)} t(x^n) - \gamma \|^2$$

$$= \| x^n - \gamma \|^2 + \frac{d^2(x^n)}{t(x^n)' t(x^n)} - \frac{2d(x^n)}{t(x^n)' t(x^n)} t(x^n)'(x^n - \gamma)$$

By definition of subgradient

$$d(\gamma) \geqq d(x^n) + t(x^n)'(\gamma - x^n)$$

Since $d(\gamma) = 0$ therefore

$$d(x^n) \leqq t(x^n)'(x^n - \gamma)$$

Collecting the above results

$$\| x^{n+1} - \gamma \|^2 \leqq \| x^n - \gamma \|^2 - \frac{d^2(x^n)}{t(x^n)' t(x^n)} .$$ ///

(2.21) Theorem [Oettli]

$$\| x^{n+1} - x^* \| \leqq \theta \| x^n - x^* \|, 0 < \theta < 1$$

Proof:

$$\| x^{n+1} - \gamma \|^2 \leqq \| x^n - \gamma \|^2 - \frac{d^2(x^n)}{t(x^n)' t(x^n)} \tag{2.22}$$

$$\leq ||x^n - \gamma||^2$$

By triangle inequality

$$||x^n - x^*|| \leq ||x^n - \gamma|| + ||\gamma - x^*||$$

$$\leq 2\ ||x^n - \gamma||$$

Since this holds for all $\gamma \varepsilon P$ including $T_P(x^n)$

$$||x^n - x^*|| \leq 2d(x^n, P)$$

Substituting x^* in 2.22

$$||x^{n+1} - x^*||^2 \leq ||x^n - x^*||^2 - \frac{d^2(x^n)}{t(x^n)'\,t(x^n)}$$

$$= ||x^n - x^*||^2 \left[1 - \frac{d^2(x^n)}{t(x^n)'\,t(x^n)} \cdot \frac{1}{||x^n - x^*||^2}\right]$$

$$\leq ||x^n - x^*||^2 \left[1 - \frac{1}{4}\frac{d^2(x^n)}{t(x^n)'\,t(x^n)} \cdot \frac{1}{d^2(x^n, P)}\right]$$

or $||x^{n+1} - x^*|| \leq ||x^n - x^*|| \left[1 - \frac{1}{4}\frac{A^2}{B^2}\right]^{\frac{1}{2}}$

where $A = \inf\limits_n \frac{d(x^n)}{d(x^n, P)}$; $B = \sup\limits_n ||t(x^n)||$

Oettli has shown that $A > 0$ and B is finite. From these observations, the result follows if we take

$$\theta = \left[1 - \frac{1}{4}\frac{A^2}{B^2}\right]^{\frac{1}{2}} \qquad ///$$

In the next chapter we show that the Generalized Merzlyakov procedure is equivalent to the Oettli procedure in as far as the sequence of possible moves made by the two procedures are concerned.

Some of the More Recent Algorithms

A limitation of subgradient optimization methods is the absence of a benchmark to compare the computational efficiency of different

algorithms. Algorithms for unconstrained optimization of smooth convex functions have the reliable method of steepest descent (Cauchy procedure), as a yardstick against which other algorithms are compared. If they are better than the Cauchy procedure, they usually deserve further consideration. In subgradient optimization, the Cauchy procedure may not work. An example where the Cauchy procedure is ineffective is reproduced below from [37]:

The function considered is

$$\max \{3x \pm 2y,\ 2x \pm 10y\}$$

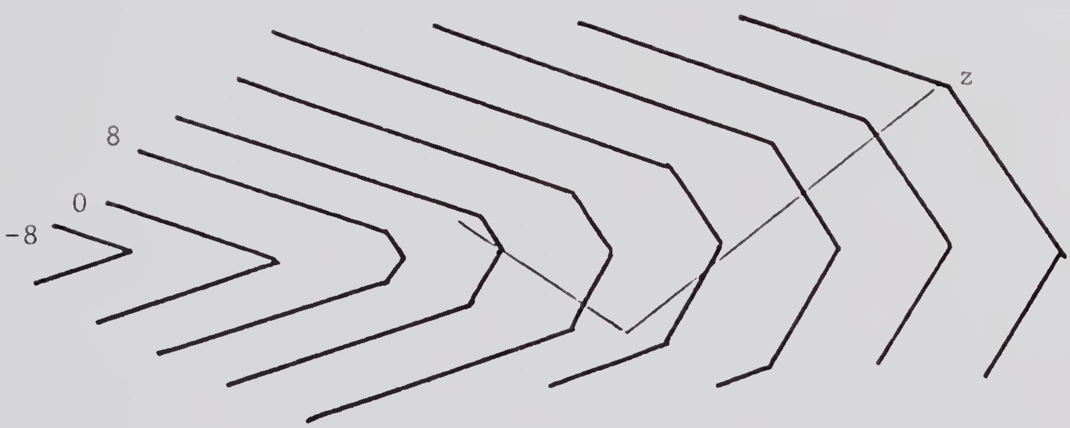


FIGURE 19

The contours of this function are sketched in Figure 19. One can calculate that if the Cauchy algorithm is started at z, the sequence converges to the origin (0,0) which is not the minimum of the function.

Some of the recent subgradient procedures can be divided into two categories. In the first category are those that claim to be at least as good as some other subgradient procedures. These algorithms claim their superiority based on empirical (computational) results. In the second category are those that claim to be reasonably effective for both differentiable and non-differntiable convex functions and claim that the algorithm is superior when applied to quadratic and convex differentiable functions.

As an example of the former we present the method of Camerini, Fratta and Maffioli [5], and as an example of the latter we present Wolfe's [38] method of conjugate subgradients for minimizing non-differentiable functions. Wolfe's algorithm closely resembles Lemarchel's method [28].

Camerini, Fratta and Maffioli. Here one adopts the following iterative scheme for minimizing $f(\cdot)$. Let

$$x^0 = 0 \text{ and}$$

$$x^{n+1} = x^n - t_n s^n \text{ where } s^n \text{ is a modified "gradient"}$$

direction defined as

$$s^{n-1} = 0 \text{ for } n = 0 \text{ and}$$

$$s^n = f'(x^n) + \beta_n s^{n-1}$$

with $f'(x^n)$ a subgradient of f at x^n and β_n a suitable scalar. s^n is equivalent to a weighted sum of preceding gradient directions and is used as an anti-jamming device. The following lemmas and theorems develop the essential elements of Camerini, Fratta and Maffioli method.

(2.23) Lemma

Let x^* and x^n be such that

$$f(x^*) \leqq f(x^n), \text{ then}$$

$$0 \leqq f(x^n) - f(x^*) = f'(x^n)'(x^* - x^n)$$

This lemma states that the negative of the subgradient makes an acute angle with the vector $(x^* - x^n)$ where x^* corresponds to an optimal solution. See Figure 20.

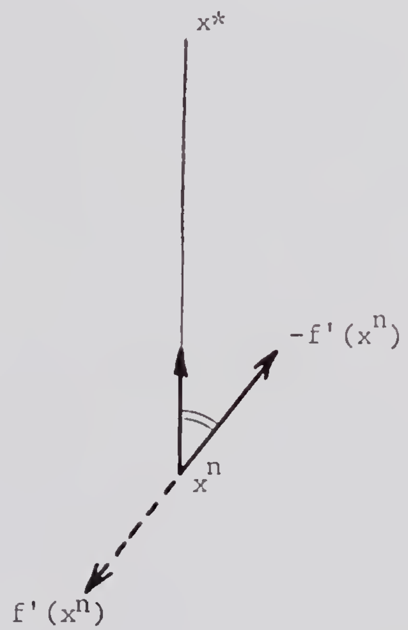


FIGURE 20

(2.24) Lemma

If for all n

$$0 < t_n \leqq \frac{f(x^n) - f(x^*)}{\|s^n\|^2} \text{ and } \beta_n \geqq 0, \text{ then}$$

$$0 \leqq f'(x^n)'(x^* - x^n) \leqq -s^{n'}(x^* - x^n)$$

Lemma 2.24 extends the subgradient property of Lemma 2.23 to $-s^n$. $-s^n$ also makes an acute angle with $(x^* - x^n)$.

(2.25) Theorem

$$\text{Let } \beta_n = \begin{cases} -\gamma_n \dfrac{s^{n-1'} f'(x^n)}{\| s^{n-1} \|^2} & \text{if } s^{n-1'} f'(x^n) < 0 \\ 0 & \text{otherwise} \end{cases}$$

with $0 \leqq \gamma_n \leqq 2$: then

$$\frac{-(x^* - x^n)' s^n}{\|s^n\|} \geqq - \frac{(x^* - x^n)' f'(x^n)}{\|f'(x^n)\|}$$

Theorem 2.25 shows that a proper choice of β_n ensures that $-s^n$ is at least as good a direction as $-f'(x^n)$ in the sense that $-s^n$ makes more of an acute angle with $(x^* - x^n)$ compared to $-f'(x^n)$. If $-f'(x^n)$ and $-s^{n-1}$ make an acute angle we take $-s^n = -f'(x^n)$. Only if $-f'(x^n)$ and $-s^{n-1}$ make an obtuse angle do we take

$$-s^n = -(f'(x^n) + \beta_n s^{n-1})$$

Figure 21 illustrates the above theorem.

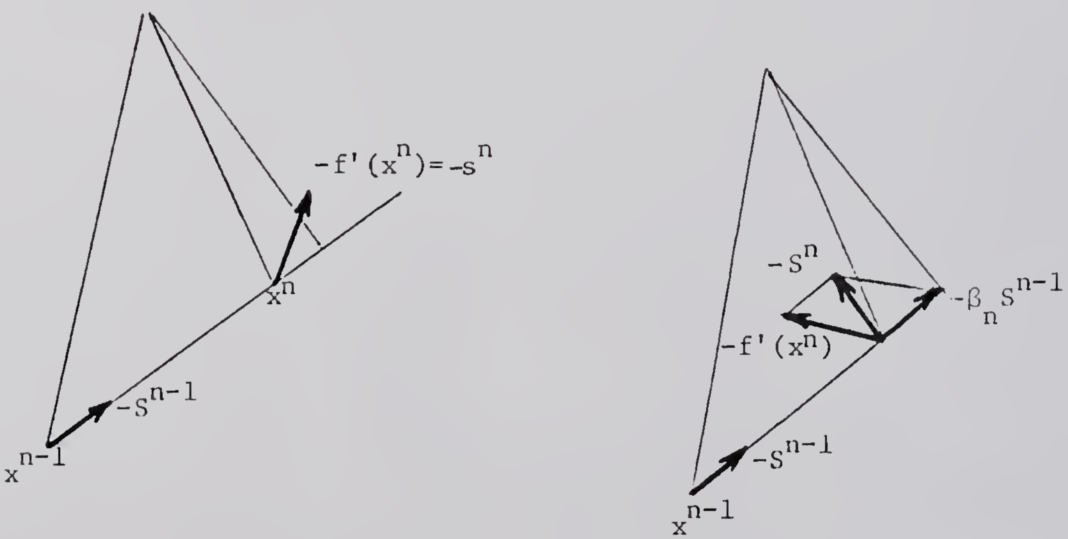


FIGURE 21

(2.26) Theorem

$$\| x^* - x^{n+1} \| < \| x^* - x^n \|$$

Proof:

$$0 < t_n \leqq \frac{f(x^n) - f(x^*)}{\| s^n \|^2}$$

$$t_n \| s^n \|^2 \leqq (f(x^n) - f(x^*)) < 2[f(x^n) - f(x^*)]$$

$$\leqq -2f'(x^n)'(x^* - x^n) \text{ by Lemma 2.24}$$

$$\leqq -2s^{n'}(x^* - x^n) \text{ by Lemma 2.25}$$

Hence

$$t_n \| s^n \|^2 + 2s^{n'}(x^* - x^n) < 0$$

or $$t_n^2 \| s^n \|^2 + 2t_n s^{n'}(x^* - x^n) < 0$$

adding $\| x^* - x^n \|^2$ to both sides gives

$$\| x^* - x^{n+1} \| < \| x^* - x^n \|$$ ///

Theorem 2.26 shows that the sequence is Fejér-monotone with respect to an optimum point x^*. Using arguments similar to those used earlier in relaxation methods, it can be shown that the sequence $\{x^n\}$ converges to x^*.

Camerini et al., use heauristic arguments to show that the best value for γ is 1.5. They then claim superiority of their method over that of Held and Karp [18] and Polyak [35] based on computational results.

Wolfe. In Wolfe's method [38] we generate a sequence of points $\{x^n\}$ of directions $\{d^n\}$, of subgradients $\{f'(x^n)\}$ and scalars $\{t^n\}$ such that for $n = 0, 1, 2, \ldots$

$$t_n \geqq 0,\ f'(x^n) \in \partial f(x^n)$$

$$x^{n+1} = x^n + t_n d^n \text{ and}$$

$$f(x^{n+1}) \leqq f(x^n).$$

At step n we have a bundle of vectors G_n which is formed by a set of rules. G_n consists of $f'(x^n)$, possibly $-d^{n-1}$ and a (possibly empty) string $f'(x^{n-1})$, $f'(x^{n-2})$, ... of limited length.

We set $d^n = -N_r G_n$ where $N_r G_n$ is a unique point in the convex hull of G_n having minimum norm. Demjanov [10] has shown that $\nabla f(x) = -N_r \partial f(x)$ where $\nabla f(x)$ is the gradient of f at x when f is smooth at x and $\partial f(x)$ is the subdifferential of f at x. As is typically the case, the stepsize, t_n, is determined by an approximate minimization of $f(x^n + td^n)$ for $t \geqq 0$. At some step, $d_n = -N_r G_n$ will be small, and in addition the last p + 1 points (for some interger p) are all close together. Then x^n will be our approximate solution.

For example consider the illustration in Wolfe's paper

$$f(x,y) = \max\ \{f_1,\ f_2,\ f_3\} \text{ where}$$

$$f_1 = -x,\ f_2 = x + y,\ f_3 = x - 2y$$

sketched in Figure 22. A subgradient of f at any (x,y) is chosen to be ∇f_i where i is the least index for which $f(x,y) = f_i$. If we start with the point (6,3), we move in the direction $-\nabla f(6,3) = (-1,-1)$ to the point (3,0). Using our prescription for choosing the direction, the only available direction at (3,0) is still (-1,-1). We must take a small step in that direction anyway even though it is not a descent direction. Then the new subgradient will be (-1,-2) forming the bundle $G_2 = \{(1,1),\ (1,-2)\}$, we find the direction $-N_r G_2 = (-1,0)$.

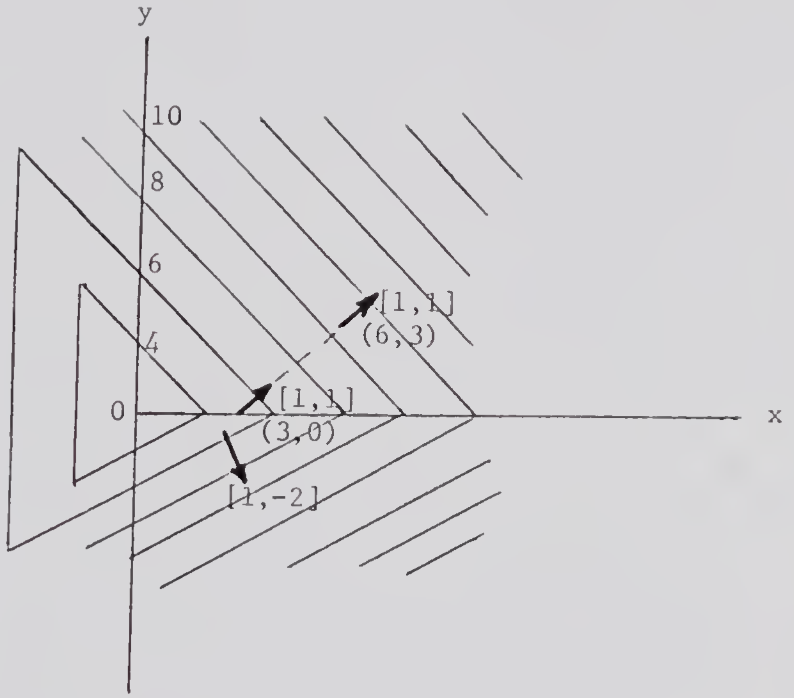


FIGURE 22

The next step in this direction takes us to the neighborhood of (0,0). At the next iteration we add (-1,0) to the bundle. Then $N_r G_3 = 0$. We "reset" and start all over again at this point. We accept the last point reached as a solution if $N_r G_n = 0$ and the last "p" points are "close enough."

For differentiable convex functions, the method is an extension of the method of conjugate gradients and terminates for quadratic functions.

CHAPTER 3

THEORETICAL RESULTS

Comparison of Generalized Merzlyakov Method and Some Typical Subgradient Methods

The equivalence of relaxation methods and minimization methods has been noted by several researchers. However, these two approaches are only mathematically equivalent and this equivalence has not been generally made explicit. On the other hand computationally relaxation methods are much simpler. It has been shown [24] that a generalized form of Merzlyakov's method subsumes most of the typical subgradient methods. In particular it was shown that the generalized Merzlyakov method can generate any move that is made by the Oettli method. In this section we draw heavily from the results of the above paper [24].

Generalized Merzlyakov Method

The Merzlyakov method is generalized [24] so as to facilitate its comparison with some of the subgradient methods. This extension greatly increases the flexibility of the original method. The following procedure allows the λ_is to be chosen based on the current point rather than being constant for a subcavity. For any x where $V(x) \neq \emptyset$, let

$$\lambda_i(x) = \begin{cases} 0 \text{ if } a_i'x + b_i > 0 \\ \geqq 0 \text{ if } a_i'x + b_i \leqq 0 \end{cases}$$

where $0 < \Sigma \lambda_i(x) \leq \beta$, $\lambda_i(x) > 0$ for some $i \varepsilon V(x)$ and at each n there is a $j \varepsilon V(x^n)$ such that

3.1 $$\frac{\lambda_j(x^n)(a_j'x^n + b_j)}{a_{j*}'x^n + b_{j*}} \geq \theta$$

where j* indexes a most violated constraint and $\theta > 0$.

For $\lambda \varepsilon (0, 2)$ and x^0 initially specified, stop at iteration n if $x^n \varepsilon P$. Otherwise, let

3.2 $$x^{n+1} = x^n - \frac{\lambda \Sigma \lambda_i(x^n)(a_i'x^n + b_i)(\Sigma \lambda_i(x^n)a_i)}{(\Sigma \lambda_i(x^n)a_i)'(\Sigma \lambda_i(x^n)a_i)}$$

We continue to assume $||a_i|| = 1$ for all i.

In the next two theorems, we show that the generalized Merzlyakov method converges with a linear rate of convergence to P.

3.3 Theorem

Let $z \varepsilon P$, and $\{x^n\}$ obtained by the generalized Merzlyakov method. Then

$$||x^{n+1} - z|| < ||x^n - z||$$

Proof:

$$||x^{n+1} - z|| = ||x^n - z||^2 - \frac{2(x^n - z)' \lambda \Sigma \lambda_i(x^n)(a_i'x^n + b_i)\Sigma \lambda_i(x^n)a_i}{(\Sigma \lambda_i(x^n)a_i)'(\Sigma \lambda_i(x^n)a_i)}$$

$$+ \frac{\lambda^2(\Sigma \lambda_i(x^n)(a_i'x^n + b_i))^2}{(\Sigma \lambda_i(x^n)a_i)'(\Sigma. \lambda_i(x^n)a_i)} .$$

The last two terms can be combined to get

$$\frac{-\lambda \Sigma \lambda_i(x^n)(a_i'x^n + b_i)}{(\Sigma \lambda_i(x^n)a_i)'(\Sigma \lambda_i(x^n)a^i)} [2\Sigma \lambda_i(x^n)a_i'(x^n - z)$$

$$-\lambda(\Sigma \lambda_i(x^n)(a_i'x^n + b_i))].$$

Now $a_i'x^n + b_i \leqq 0$ whenever $\lambda_i(x^n) > 0$ and $a_i'x^n + b_i < 0$ for some $\lambda_i(x^n) > 0$ and thus the first bracketed term is positive.

Since $z \varepsilon P$ and $a_i'z \geqq b_i$, then

$$-\Sigma \lambda_i(x^n)a_i'z \leqq \Sigma \lambda_i(x^n)b_i$$

Thus the second bracketed term is less than or equal to

$$(2 - \lambda) \Sigma \lambda_i(x^n)(a_i'x^n + b_i)$$

and this term is strictly negative.

Thus

$$\| x^{n+1} - z \| < \| x^n - z \| \qquad ///$$

Thus we have established that the sequence is Fejér-monotone with respect to P. We next show that the procedure gives points which converge linearly to P. Let

$$d(x^n, P) = \inf_{z \varepsilon P} \| x^n - z \| .$$

3.4 Theorem

Either the generalized Merzlyakov procedure terminates with a point of P or it generates a sequence which converges to a point $x^* \varepsilon P$.

Furthermore, the convergence rate is linear.

Proof: Assume $x^n \notin P$ and z^n is the unique closest point of P to x^n.

Then

$$d^2(x^{n+1}, P) \leqq \| x^{n+1} - z^n \|^2.$$

After expanding $\| x^{n+1} - z^n \|^2$ we get

$$d^2(x^{n+1}, P) \leqq \| x^n - z^n \|^2 - \frac{\lambda(2-\lambda)[\Sigma \lambda_i(x^n)a_i'x^n + b_i)]^2}{(\Sigma \lambda_i(x^n)a_i)'(\Sigma \lambda_i(x^n)a_i)}$$

$$= d^2(x^n, P) - \frac{\lambda(2-\lambda)[\Sigma \lambda_i(x^n)(a_i'x^n + b_i)]^2}{(\Sigma \lambda_i(x^n)a_i)'(\Sigma \lambda_i(x^n)a_i)} .$$

By assumption

$$[\Sigma\ \lambda_i(x^n)(a_i'x^n + b_i)]^2 \geqq [\lambda_i(x^n)(a_i'x^n + b_i)]^2$$

$$\geqq \theta^2(a_{j*}'x^n + b_{j*})^2 \text{ for some } i \in V(x^n)$$

From Hoffman [21] we have that there is a $\mu > 0$

$$\mu d(x^n, P) \leqq -a_{j*}'x^n - b_{j*}$$

Thus

$$d^2(x^{n+1}, P) \leqq d^2(x^n, P) - \frac{\lambda(2-\lambda)\mu^2\theta^2 d^2(x^n, P)}{(\Sigma\ \lambda_i(x^n)a_i)'(\Sigma\ \lambda_i(x^n)a_i)}$$

Finally since $\Sigma\ \lambda_i(x^n) \leqq \beta$ and $\|a_i\| = 1$ for all i, we have

$$(\Sigma\ \lambda_i(x^n)a_i)'(\Sigma\ \lambda_i(x^n)a_i) \leqq \beta^2$$

And we get

$$d^2(x^{n+1}, P) \leqq [1 - \lambda(2-\lambda)\theta^2\mu^2/\beta^2]d^2(x^n, P) \equiv \gamma^2 d^2(x^n, P)$$

Then $0 \leqq \gamma < 1$ and $d(x^n, P) \leqq \gamma^n d(x^0, P)$.

Case 1: If the sequence does not terminate $\lim_{n\to\infty} d(x^n, P) = 0$ and

Motzkin and Shoenberg [31] have shown that the sequence converges to a point x* in the boundary of P for $0 < \lambda < 2$.

Case 2: If the sequence terminates

$$\|x^{n+1} - z\| < \|x^n - z\| \text{ for all } z \in P \text{ (Theorem 3.3).}$$

Also $\|x^* - z^n\| < \|x^n - z^n\|$ where z^n is the closest point to P from x^n.

Thus

$$\begin{aligned}\|x^* - x^n\| &\leqq \|x^* - z^n\| + \|z^n - x^n\| \\ &< 2\|x^n - z^n\| = 2d(x^n, P) \\ &\leqq 2\gamma^n d(x^0, P)\end{aligned}$$

where x* is the last point in the terminating sequence. ///

We can further generalize the Merzlyakov method and Theorems 3.3 and 3.4 by replacing λ with λ_n where $0 < b_i \leq \lambda_n \leq b_2 < 2$ for all n. In the remainder of the discussion we usually mean $\lambda_n = \lambda$ for all n but do not have to restrict ourselves in this manner.

Further Developments of Oettli Method

Oettli's minimization method was discussed in the last chapter. We wish to show the equivalence of Generalized Merzlyakov method and Oettli's method in the next section but before that we need to develop some properties of Oettli's method to facilitate this comparison. Again most of the theorems and proofs in this section are taken from the paper [24] referred to earlier.

We represent the subdifferential of a convex function f at x by $\partial f(x)$and a subgradient of f at x by $f'(x)$ i.e., $f'(x) \in \partial f(x)$.

To recapitulate the Oettli procedure, we form the sequence $\{x^n\}$ as follows:

3.6 $$x^{n+1} = x^n - \frac{d(x^n)t(x^n)}{t(x^n)'t(x^n)}$$

where $t(x^n)$ is any subgradient of $d(\cdot)$ at x^n.

3.7 Lemma

For any norm $p(\cdot)$ on R^m

$$\partial p(x) = \partial p(0) \cap \{p'(x)\colon p(x) = p'(x)'x\}$$

Proof:

We will show that if $p'(x) \in \partial p(x)$then the following ralation holds

$$\partial p(x) = \{p'(x)\colon p(y) \geqq p'(x)'y \text{ for all } y\} \cap$$
$$\{p'(x)\colon p(x) = p'(x)'x\}$$

Conversely if the above relation holds, we will show $p'(x) \in \partial p(x)$.

By the definition of a subgradient $p'(x) \varepsilon \partial p(x)$ if

$$p(y) \geqq p(x) + p'(x)'(y - x) \text{ for all } y.$$

Also

$$p(\lambda x) - p(x) \geqq p'(x)'(\lambda x - x) \text{ for all scalars } \lambda.$$

Since $p(\cdot)$ is a norm

$$p(\lambda x) = |\lambda| p(x)$$

For $\lambda > 1$ $\quad p(x) \geqq p'(x)'x$

$0 < \lambda < 1$ $\quad p(x) \leqq p'(x)'x$

Thus $p(x) = p'(x)'x$.

From the subgradient inequality

$p(y) \geqq p(x) + p'(x)'(y - x)$ by substituting for $p(x)$ we get

$p(y) \geqq p'(x)'y$ for all y. Hence

$$\partial p(x) = \{p'(x)\colon p(y) \geqq p'(x)'y \text{ for all } y\} \cap \{p'(x)\colon p(x) = p'(x)'x\}.$$

Conversely, if $p(y) \geqq p'(x)'y$, for all y and $p(x) = p'(x)'x$, we get by subtraction

$$p(y) - p(x) \geqq p'(x)'y - p(x) = p'(x)(y - x)$$

and $p'(x) \varepsilon \partial p(x)$.

Finally

$$\partial p(0) = \{p'(0)\colon p(y) \geqq p'(0)'y, \text{ for all } y\} \cap R^m$$

$$= \{p'(0)\colon p(y) \geqq p'(0)'y, \text{ for all } y\}. \qquad \boxed{///}$$

To find the subgradients of $d(\cdot)$ in terms of subgradients of the composite function $p(\ell^+(x))$ we use the following chain rule given by Oettli.

3.8 Lemma

Let $p(\cdot)$ be an isotone norm and $p'(\ell^+(z)) \geqq 0$, then

3.9 $\qquad d'(z) = \ell^{+'}(z)\, p'(\ell^+(z))$

Proof:

$$\ell^+(z) - \ell^+(z^0) \geqq (z - z^0)'\ell^{+'}(z^0)$$

multiplying both sides by $p'(\ell^+(x))$ where $p'(\ell^+(x)) \geqq 0$

$$(\ell^+(z) - \ell^+(z^0))p'(\ell^+(z^0)) \geqq (z - z^0)'\ell^{+'}(z^0)p'(\ell^+(z^0))$$

Using the definition of subgradient for $p'(\ell^+(z^0))$

$$p(\ell^+(z)) - p(\ell^+(z^0)) \geq (\ell^+(z) - \ell^+(z^0))'p'(\ell^+(z^0))$$

$$\geq (z - z^0)'\ell^{+'}(z^0)p'(\ell^+(z^0))$$

$$d(z) - d(z^0) \geqq (z - z^0)'\ell^{+'}(z^0)p'(\ell^+(z^0)).$$

But $d(z) - d(z^0) \geqq (z - z^0)'d'(z^0)$

hence $d'(z^0) = \ell^{+'}(z^0)p'(\ell^+(z^0))$.

The subgradient of $\ell_i^+(x)$ may be found using the following well known result

3.10 $$\partial\ell_i^+(x) = \{-\lambda a_i : \lambda = 0 \text{ if } a_i'x + b_i > 0$$
$$\lambda = 1 \text{ if } a_i'x + b_i < 0 \text{ and}$$
$$\lambda \in [0,1] \text{ if } a_i'x + b_i = 0\}$$

The subdifferential of $\ell^+(x)$ can in turn be found from

3.11 $$\partial\ell^+(x) = \{(h_1, \ldots, h_m): h_i \in \partial\ell_i^+(x)\}$$

Generalized Merzlyakov method and Oettli method

With the results developed so far in the previous sections we are now in a position to develop an expression for equation 3.6 in terms of the parameters of the generalized Merzlyakov method.

3.12 Theorem

Let $x \in R^k$, $x \notin P$ and $t(x)$ be a subgradient of $d(\cdot)$ at x formed by using the chain rule of equation 3.9. Then

$$\frac{d(x)t(x)}{t(x)'t(x)} = \frac{[\Sigma\lambda_i(a_i'x + b_i)]\Sigma\lambda_i a_i}{(\Sigma\lambda_i a_i)'(\Sigma\lambda_i a_i)}$$

for some λ_i's where $\lambda_i \geqq 0$, $\lambda_i = 0$ if $a_i'x + b_i > 0$, and $\Sigma\lambda_i > 0$.

Proof:

By equation 3.9 we have

$$t(x) = \ell^{+'}(x)p'(\ell^+(x)) \text{ for some } \ell^{+'}(x) \in \partial\ell^+(x)$$

and $p'(\ell^+(x)) \in \partial p(\ell^+(x))$.

By equations 3.10 and 3.11

$$\ell^{+'}(x) = -(\eta_1 a_1, \eta_2 a_2, \ldots, \eta_m a_m) \text{ with}$$

$$\eta_i = \begin{cases} 0 \text{ if } a_i'x + b_i > 0 \\ 1 \text{ if } a_i'x + b_i < 0 \\ \in [0,1] \text{ if } a_i'x + b_i = 0 \end{cases}$$

Also from the proof of Lemma 3.7

$$d(x) = p'(\ell^+(x))'\ell^+(x) \text{ where}$$

$$\ell^+(x) = \begin{pmatrix} \eta_1(a_i'x + b_i) \\ \vdots \\ \vdots \\ \eta_m(a_m'x + b_m) \end{pmatrix}$$

Let

$h \equiv p'(\ell^+(x))$. In Oettli's method we need $h \geqq 0$. Substituting these last few expressions into $\frac{d(x)t(x)}{t(x)'t(x)}$ gives

$$\frac{d(x)t(x)}{t(x)'t(x)} = \frac{[\Sigma\lambda_i(a_i'x + b_i)]\Sigma\lambda_i a_i}{(\Sigma\lambda_i a_i)'(\Sigma\lambda_i a_i)}$$

with $\lambda_i = \eta_i h_i \geqq 0$. We also have $-\Sigma\lambda_i(a_i'x + b_i) = p(\ell^+(x)) > 0$ since $x \notin P$ and $\Sigma\lambda_i > 0$. ///

We have thus established a close similarity between the two methods. We have however yet to establish the two other requirements viz $\Sigma\lambda_i \leqq \beta$ and that condition 3.1 is satisfied. First we note that

since the subdifferential of $p(\cdot)$ is compact $\Sigma\lambda_i$ is bounded above. Hence the condition $\Sigma\lambda_i \leqq \beta$ is satisfied. Now to show that condition 3.1 is satisfied, we see on the lines of the proof of Theorem 3.12 that

$$d(x) = p(\ell^+(x)) = p'(\ell^+(x))'\ell^+(x) \equiv h'(\ell^+(x))$$

$$= -\Sigma\eta_i h_i(a_i'x + b_i) \equiv -\Sigma\lambda_i(x)(a_i'x + b_i)$$

where $\lambda_i(x) = \eta_i h_i \geqq 0$.

Hence

$$-\Sigma\lambda_i(x)(a_i'x + b_i) = p(\ell^+(x))$$

and $$m \max_i \lambda_i(x)(-a_i'x - b_i) \geqq p(\ell^+(x))$$

Also $\ell^+(x) \geqq \ell^+_{j*}(x)e_{j*}$ where $j*$ is the most violated constraint and e_{j*} is a unit vector with a 1 in position $j*$. Since $p(\cdot)$ is monotonic

$$p(\ell^+(x)) \geqq p(\ell^+_{j*}(x)e_{j*})$$

$$= -(a_{j*}'x + b_{j*})p(e_{j*})$$

Thus

$$m \max_i \lambda_i(x)(-a_i'x - b_i) \geqq -(a_{j*}'x + b_{j*})p(e_{j*})$$

$$\geqq -(a_{j*}'x + b_{j*}) \min_i p(e_i)$$

Hence

$$\frac{\lambda_i(x)(a_i'x + b_i)}{(a_{j*}'x + b_{j*})} \geqq \frac{\min_i p(e_i)}{m} \equiv \theta > 0$$

where $i \in V(x)$ maximizes the left hand side.

If we allow $\lambda = 1$ in the generalized Merzlyakov method, we find from the above analysis that any move made by the Oettli procedure

can also be made by the generalized Merzlyakov method. However, the reverse is not true. This can be seen if we let $\lambda \neq 1$ and take the case where x violates only one constraint. Thus the Oettli procedure is strictly subsumed by the generalized Merzlyakov method. Computationally the relaxation method is very simple to implement since the directions of movement can be easily computed. Also it is a very flexible procedure since all that is required to change the direction of movement and the step length is to change the weights $\lambda_i(x)$. On the other hand the Oettli method requires a subgradient to be computed at each point which is not a trivial calculation.

In an earlier paragraph we indicated that the Merzlyakov procedure could be further generalized by allowing λ to vary with each iteration. However this would require the additional condition $0 < b_i \leqq \lambda_n \leqq b_2 < 2$. Oettli has recently generalized his method [33] to incorporate the above feature. His generalized method requires that $\lambda_n \in (0, 2)$ for all n with the additional stipulation

$$\Sigma\lambda_n(2 - \lambda_n) = + \infty.$$

A second aspect in which the two methods differ is that in the generalized Merzlyakov method we could use a different set of λ_i's at each iteration. This could be introduced into the Oettli method by considering different norms at each iteration.

Comparison With Some Other Subgradient Methods

In this section we show the versatility of the generalized Merzlyakov method by showing that some of the other typical subgradient procedures are strictly subsumed by the generalized Merzlyakov method.

We consider the methods of Polyak ([34], [35]), Eremin [12] and Held, Wolfe and Crowder [19]. These subgradient methods are capable of tackling more general functions, however we compare them when the objective is merely to find a point of a polyhedron P. There are a number of ways of defining the function $f(\cdot)$ such that when we minimize it we get $x^* \varepsilon P$. A typical choice is

3.13 $$f(x^n) = \max_{i=1,\ldots,m} -(a_i' x^n + b_i).$$

We will show that with f thus defined the subgradient methods are subsumed by the generalized Merzlyakov method. The above subgradient methods can be collectively described by the following sequence:

$$x^{n+1} = x^n - \lambda_n \frac{[f(x^n) - \bar{f}]}{\|f'(x^n)\|^2} f'(x^n)$$

where $\bar{f}$ is an estimate of $f(x^*)$ the optimum value of f. If $f(x^n)$ assumes its value for a unique i in the relation (3.13) then the subgradient at x^n, $f'(x^n) = a_i$. If there are ties, $f'(x^n)$ can be taken to be a convex combination of the maximizing a_is. Let $\emptyset(x^n)$ represent the set of indices $i \varepsilon \{1, \ldots, m\}$ for which

$$f(x^n) = \max_{i=1,\ldots,m} (-a_i' x^n - b_i). \quad \text{Then}$$

$$f'(x^n) = \Sigma\lambda_i(x^n) a_i \quad \text{with } \Sigma\lambda_i(x^n) = 1$$

$\lambda_i(x^n) \geqq 0$ and $\lambda_i(x^n) = 0$ for $i \notin \emptyset(x^n)$.

Hence

$$x^{n+1} = x^n - \frac{\lambda n (f(x^n) - \bar{f})}{\|f'(x^n)\|^2} f'(x^n)$$

$$= x^n - \lambda_n \frac{(\Sigma\lambda_i(x^n)(a_i' x^n + b_i) - \bar{f})(\Sigma\lambda_i(x^n) a_i)}{(\Sigma\lambda_i(x^n) a_i)'(\Sigma\lambda_i(x^n) a_i)}$$

with $\lambda_i(x^n) \geq 0$, $\Sigma\lambda_i(x^n) = 1$ and $\lambda_i(x^n) = 0$ for $i \notin \emptyset(x^n)$.

This can be considered equivalent to finding a point of $\tilde{P}$ where

$$\tilde{P} = \{x: a_i x + b_i - \overline{f} \geq 0, \; i = 1, \ldots, m\}.$$

If $\overline{f} \geq 0$, the procedure gives a point of P. Thus if $\tilde{P} \neq \emptyset$ the subgradient procedures mentioned in this section are special cases of the generalized Merzlyakov method.

An Algorithm Using Relaxation Method

Motivation

We have seen that with relaxation parameter of $\lambda = 1$ we can get the best convergence rate for a fixed μ (see Figure 14). Also a higher μ leads to better convergence. We wish to combine both these features in our algorithm. We also derive motivation from Polyak's [34] and Oettli's procedure [32]. In this section we give a procedure for finding a point of an arbitrary polyhedron P and later apply it to the specific problem of solving large linear programs.

Given a polyhedron P

$$P = \{x \in R^k: a_i'x + b_i \geqq 0, \; i \in I\}.$$

We assume $P \neq \emptyset$ and $I \neq \emptyset$ and finite. Goffin [15] has shown that when P is full dimensioned finite convergence is assured for a range of relaxation parameters λ depending on the obtuseness index of the polyhedron. When P is not full dimensioned, we can partition it to subsets M and C, such that M is full dimensioned, and then devise an iterative procedure relative to these subsets so as to take advantage of the full dimensionality. Even though finite convergence is not assured, we can hope to get a better convergence rate with a proper choice of the relaxation parameters. There is considerable flexibility

in the manner M and C are chosen. One important consideration is that it should be easy to project on to C. The algorithm is in two phases.

In Phase I we use an iterative scheme on the sets M and C which drives us close to P. We do this by starting with an arbitrary point in C. We then move in the direction of M. This movement could be a projection onto M, or a reflection or under or over relaxation, depending on the value of the relaxation parameter λ. From the new point thus obtained we project on to C. We show that the sequence thus constructed converges to P with a linear rate of convergence. In our algorithm however, this iterative procedure is continued merely to get close to P. In this special case when $C = R^k$, Phase I becomes the relaxation procedure of Agmon.

Once we get close to P we switch to Phase II. In Phase II we consider the constraints of the set P and apply the Generalized Merzlyakov method. The motivation for use of the Generalized Merzlyakov method at this stage is the fact that the set of violated constraints as we approach P are precisely the set of violated constraints at the limit point and under these circumstances, the Generalized Merzlyakov method can be very effective.

We first describe the procedure for finding a point of an arbitrary polyhedron P.

<u>The Algorithm</u>

<u>Phase I</u>.

Let $z^0 \varepsilon$ C-P

If $z^n \varepsilon$ P stop, otherwise define

$$z^{n+1} = T_C\{z^n + \lambda d_M(z^n)a_{i_n}\} \equiv T_C(s^n)$$

where $0 < \lambda < 2$ and a_{i_n} corresponds to the most violated halfspace of M at z^n and

3.14 $$s^n \equiv z^n + \lambda d_M(z^n)a_{i_n}.$$

We will show that the sequence $\{z^n\}$ is Fejér-monotone with respect to P.

3.15 Lemma [Goffin]

Let x, y, t be points of R^k and let

$x' = x + \lambda(t - x)$ where $\lambda \in R$. Then

$$\|x' - y\|^2 = \|x - y\|^2 - \lambda(2 - \lambda)\|t - x\|^2 + 2\lambda(t-y)'(t-x)$$

Proof:

$$\|x' - y\|^2 = \|x + \lambda(t - x) - y\|^2$$
$$= \|x - y\|^2 + [2(x - y) + \lambda(t - x)]'\lambda(t - x)$$
$$= \|x - y\|^2 - \lambda(2 - \lambda)\|t - x\|^2 + \lambda[2(x - y) + 2(t-x)]'(t-x)$$
$$= \|x - y\|^2 - \lambda(2 - \lambda)\|t - x\|^2 + 2\lambda(t - y)'(t - x). \qquad ///$$

3.16 Lemma

$$\|s^n - u\|^2 = \|z^n - u\|^2 - \lambda d_M(z^n)[(2 - \lambda)d_M(z^n) - 2a'_{i_n}(t^n - u)]$$

for $u \in M$.

Proof:

Substitute s^n for x', z^n for x and t^n for t, where

$t^n = z^n + d_M(z^n)a_{i_n}$. Then

$$\|s^n - u\|^2 = \|z^n - u\|^2 - \lambda(2 - \lambda)\|t^n - z^n\|^2 + 2\lambda(t^n - u)'(t^n - z^n)$$
$$= \|z^n - u\|^2 - \lambda(2 - \lambda)d_M^2(z^n) + 2\lambda d_M(z^n)a'_{i_n}(t^n - u)$$
$$= \|z^n - u\|^2 - \lambda d_M(z^n)[(2 - \lambda)d_M(z^n) - 2a'_{i_n}(t^n - u)]. \qquad ///$$

3.17 Theorem

Let M be a non-empty polyhedron of R^k. Then the sequence of points generated by equation (3.14) has the following property:

$$||s^n - u|| \leqq ||z^n - u|| \text{ for all } u \in M.$$

Proof:

If $u \in M$, $u \in H_{i_n}$. Hence $a'_{i_n}(t^n - u) \leqq 0$ for all $u \in M$. By lemma 3.16

$$||s^n - u|| \leqq ||z^n - u||.$$ ///

3.18 Theorem

Let P be a non-empty polyhedron of R^k. Then the sequence of points $\{z^n\}$ generated by equation (3.14) is Fejér-monotone with respect to P, in fact

$$||z^{n+1} - u|| < ||z^n - u|| \text{ for all } u \in P.$$

Proof:

Let $u \in P$ and let b be the unique point obtained by projecting u onto the line formed by extending s^n, z^{n+1} (Figure 23). The angle formed by s^n, b, u is a right angle and for the right triangle s^n, b, u.

$$||s^n - a||^2 + ||a - u||^2 = ||s^n - u||^2$$

$$||s^n - z^{n+1}||^2 + ||z^{n+1} - a||^2 + 2||s^u - z^{n+1}|| \, ||z^{n+1} - a|| + ||a - u||^2 = ||s^n - u||^2$$

or

$$||s^n - z^{n+1}||^2 + ||z^{n+1} - u||^2 + 2||s^n - z^{n+1}|| \, ||z^{n+1} - a|| = ||s^n - u||^2$$

or

$$||z^{n+1} - u||^2 < ||s^n - u||^2.$$

The strict inequality arises from the fact that s^n and z^{n+1} are distinct or else $z^{n+1} \varepsilon P$. But by Theorem (3.17)

$$\|s^n - u\| \leqq \|z^n - u\| \quad \text{hence}$$

$$\|z^{n+1} - u\| < \|z^n - u\| \qquad ///$$

In the next theorem we show that the sequence $\{z^n\}$ converges linearly to P.

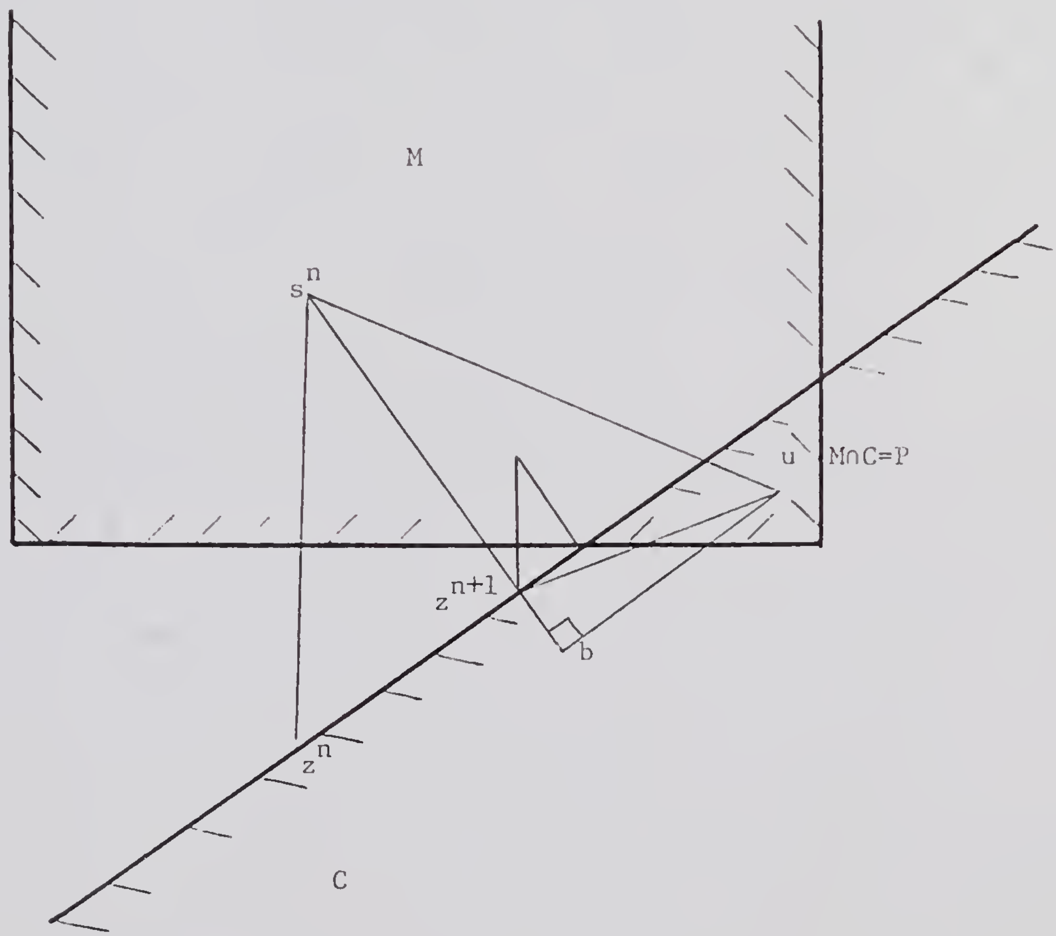


FIGURE 23

3.19 Theorem

The sequence of points $\{z^n\}$ generated by the above procedure converges finitely or infintely to a point of P. Furthermore the convergence rate is linear.

Proof:

$||z^{n+1} - T_P(z^{n+1})||^2 = ||z^{n+1} - u||^2$ for a unique $u \varepsilon P$ (the closest point of z^{n+1} to P). Thus

$$||z^{n+1} - u||^2 = ||T_C\{z^n + \lambda d_M(z^n)a_{i_n}\} - T_C(u)||^2$$
$$\leqq ||z^n - u||^2 - \lambda d_M(z^n)[2a'_{i_n}(z^n-u)-\lambda d_M(z^n)]$$

since the projection map satisfies the Lipschitz condition with $c = 1$. Continuing

$$||z^{n+1} - u||^2 \leqq ||z^n - u||^2 - \lambda d_M(z^n)[2d_M(z^n) - \lambda d_M(z^n)]$$

since $\quad a'_{i_n}(z^n - u) = d_M(z^n)$ hence

$$||z^{n+1} - u||^2 \leqq d^2(z^n, P)[1 - \lambda(2 - \lambda)\mu^{*2}] \text{ by Lemma 2.12}$$
$$= \theta^2 d^2(z^n, P).$$

We get

$$d(z^{n+1}, P) \leqq \theta d(z^n, P) \text{ where}$$
$$\theta = [1 - \lambda(2 - \lambda)\mu^{*2}]^{1/2}$$

and $\theta \varepsilon [0, 1)$.

Hence

$d(z^n, P) \leqq \theta^n d(z^0, P)$ and if the sequence is infinite

$$\lim_{n \to +\infty} d(z^n, P) = 0$$

Using Theorem [2.4] and by the continuity of the distance function this implies $\{z^n\}$ converges to z^* in the boundary of P.

If the sequence terminates at z^*, then

$$\| z^* - T_P(z^n) \| \leqq \| z^n - T_P(z^n) \| = d(z^n, P)$$

by Fejér-monotonicity of the sequence. Hence z^* and z^n both belong to the ball $B_{d(z^n, P)}(T_P(z^n))$, and

$$\| z^* - z^n \| \leqq 2d(z^n, P) \leqq 2\theta^n d(z^0, P). \qquad ///$$

Phase II. When the last k points are within ε of each other, we switch to the Generalized Merzlyakov procedure. This is because if we are near the limit point, the only active constraints would be the tight constraints at the limit point of the sequence. In such a situation the Generalized Merzlyakov method can be very effective with a proper choice of λ and $\lambda_j(x)$. We recapitulate that in Merzlyakov's procedure (which is a special case of Generalized Merzlyakov method) the sequence is generated as follows:

$$x^{n+1} = x^n - \frac{\lambda[\Sigma\lambda_i(j, V)(a_i'x^n + b_i)][\Sigma\lambda_i(j, V)a_i]}{[\Sigma\lambda_i(j, V)a_i]'[\Sigma\lambda_i(j, V)a_i]}$$

where $x^n \varepsilon S_V^j$.

If we assume that we know the set of constraints satisfied with equality by $T_P(x^n)$, we can generate the halfspace

$$H^* = \{x \varepsilon R^n : (T_P(x^n) - x^n)'(x - T_P(x^n)) \geqq 0\}$$

which gives convergence to $T_P(x^n)$ in exactly one step with $\lambda = 1$. Suppose we number the set of active constraints at $T_P(x^n)$ by $I^* = \{1, \ldots, p\}$. Let $A^* \equiv (a_1, \ldots, a_p)'$ be the p x k matrix formed by interior normals of the halfspaces H_i for $i \varepsilon I^*$. Let $b^* = (b^1, \ldots, b^p)'$. Define

$$(\lambda^1, \ldots, \lambda^p)' = -(A^*A^{*\prime})^{-1}(A^*x^n + b^*).$$

With this set of $\lambda_i(j, V)$'s using Merzlyakov method, Goffin [15] has shown that convergence to $T_p(x^n)$ is obtained in one step. However, this requires a comparably larger amount of computation and finding $(A^*A^{*\prime})^{-1}$ is not really practical on large problems. In this study we have therefore concentrated on the Generalized Merzlyakov Method and attempted to find what would constitute a favorable set for λ and $\lambda_j(x)$.

Application to Solving Large Linear Programs

We now consider how the procedure could be used to solve large linear programs.

Consider the LP

$$\max c'x$$
$$\text{s.t. } Ax \leqq b$$
$$x \geqq 0$$

and its dual

$$\min \pi'b$$
$$\text{s.t. } A'\pi \geqq c$$
$$\pi \geqq 0.$$

By stong duality, an optimal π and x satisfy

$b'\pi - c'x \leqq 0.$ Let

$$C = \{\binom{x}{\pi} : b'\pi - c'x \leqq 0\}$$

$$M = \{\binom{x}{\pi} : Ax \leqq b,\ -A'\pi \leqq -c,\ x \geqq 0,\ \pi \geqq 0\}$$

and $P = M \cap C.$

This is only one of the ways of partitioning the constraint of P. There is in fact great flexibility in the choice of C and M. The advantage of splitting C and M is indicated above is

(a) M has a special structure and can be decomposed along π and x. This leads to saving in storage as well as easier arithmetic.

(b) C contains only the coupling constraints and is easy to project to.

However, there are other ways of obtaining P. Another choice could be to let C have only the non-negativity constraints. Again the advantage of such a construction would be ease of projection to C. Still another choice is for C to take the form

$$C = \{\binom{x}{\pi}: b'\pi - c'x \leqq 0,\ x \geqq 0,\ \pi \geqq 0\}.$$

There is user discrimination in this algorithm in the choice of M and C, λ and Z^0 in Phase I, and λ and weights $\lambda_j(x)$ in Phase II. We have already commented on the possible strategies for selecting M and C. We give below strategies for selecting the other parameters. We will first discuss the parameters for Phase I.

(1) $\underline{\lambda}$: If $0 < \lambda \leqq 1$ convergence may be slow if some of the "angles" of M are very acute (small obtuseness index of P). In this case increasing λ to near 2 will have the effect of opening the angles and accelerating the convergence. It therefore appears to be a good strategy to have λ approximately equal to 2 in Phase I.

(2) $\underline{Z^0}$: In line with the general observation for relaxation methods that any a priori information on the location of the solution could be used to advantage, if we have to solve a linear program with only slightly varying data, we could take advantage of this feature. However, in general when we do not have any such prior information, the choice of Z^0 could be dictated by the structure of C, so that we can start with a feasible point of C.

Now for the parameters of Phase II

(1) $\underline{\lambda}$: As a general guideline $1 < \lambda < 2$ may be chosen as the parameter for Phase II as was done in Phase I. However, Table 3 and Figure 24 show the sequence obtained with $\lambda = 1.5$ and $\lambda = .75$ and illustrate that in some cases choice of $0 < \lambda < 1$ may give better results.

(2) $\underline{\lambda_i(j, V)}$: In Merzlyakov method the choice of weights $\lambda_i(j, V)$ is crucial and in a large measure dictates the convergence of the sequence. It may be noted that $\lambda_i(j, V)$'s determine both the direction as well as the step size. Suppose a_i, $i \varepsilon K$ represent the set of violated constraints in subcavity S_V^j where $x^n \varepsilon S_V^j$. We can solve the quadratic problem of finding the point of minimum norm in convex hull of the finite point set a_i, $i \varepsilon K$ to get the weights $\lambda_i(j, V)$. Such a direction locally has a great deal of merit. However the computational effort is too great to merit its consideration, since at each iteration we have to solve a quadratic program. Also it may not always lead to the best set of weights as shown by the counterexample given in the next paragraph. Instead we specify $\lambda_i(j, V) = \frac{1}{|K|}$, where $|K|$ represents the number of constraints violated at x^n. When $|K| = 2$ the direction obtained coincides with the vector of minimum norm.

The strategy suggested by us is not the best in all cases as shown by the following counterexample:

Consider $x_1 \geqq 1$

$$x_2 \geqq 2$$

$$x^1 = \binom{0}{0}$$

TABLE 3

Case 1: $\lambda = .75$ $\lambda_i(j, V) = 1$ for all $i \varepsilon V$.

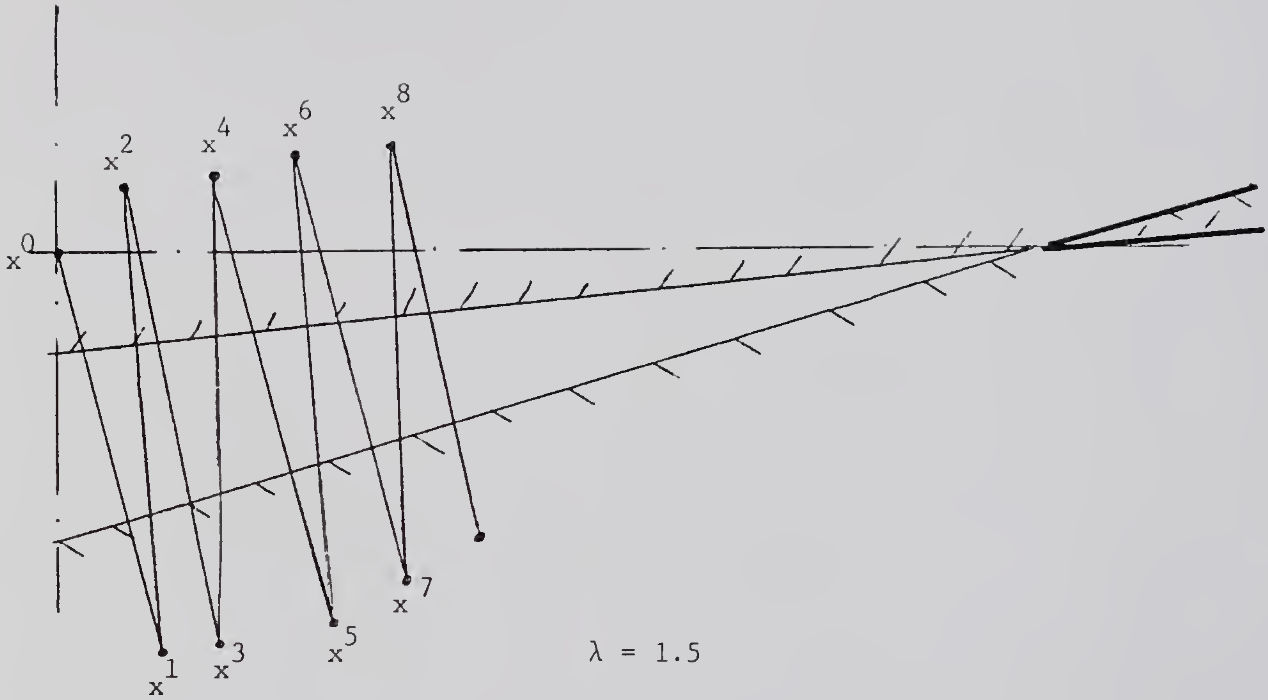
n	x_1^n	x_2^n	$\Sigma\lambda_i(j, V) \cdot (a_i'x^n + b_i)$	$\Sigma\lambda_i(j, V)a_i$	$[\Sigma\lambda_i(j, V)a_i]' \cdot [\Sigma\lambda_i(j, V)a_i]$
0	0.000	0.000	-2.873	$\binom{.287}{-.958}$	1
1	.619	-2.064	-1.838	$\binom{.188}{.037}$	.037
2	7.622	- .686	- .459	$\binom{.188}{.037}$	.037
3	9.371	- .342	- .273	$\binom{-.099}{.995}$	1
4	9.351	- .138	- .125	$\binom{.188}{.037}$	.037
5	9.827	- .044	- .032	$\binom{.188}{.037}$	.037
6	9.949	- .020	- .010	$\binom{-.099}{.995}$	1
7	9.948	- .013	- .008	$\binom{.188}{.037}$	.037
8	9.978	- .007	- .002	$\binom{.287}{-.958}$	1

Case 2: $\lambda = 1.5$ $\lambda_i(j, V) = 1$ for all $i \varepsilon V$.

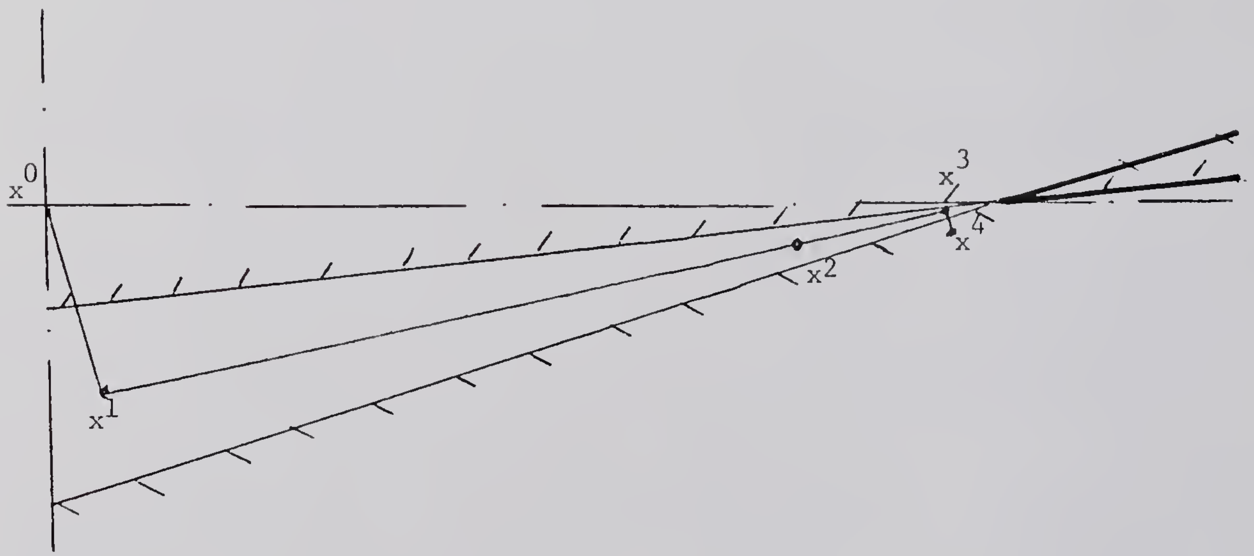
n	x_1^n	x_2^n	$\Sigma\lambda_i(j, V) \cdot (a_i'x^n + b_i)$	$\Sigma\lambda_i(j, V)a_i$	$[\Sigma\lambda_i(j, V)a_i]' \cdot [\Sigma\lambda_i(j, V)a_i]$
0	0.000	0.000	-2.873	$\binom{.287}{-.958}$	1
1	1.237	-4.128	-3.235	$\binom{-.099}{.995}$	1
2	.757	.700	-3.326	$\binom{.287}{-.958}$	1
3	2.189	-4.079	-3.280	$\binom{-.099}{.995}$	1
4	1.702	.816	-3.166	$\binom{.287}{-.958}$	1
5	3.065	-3.733	-3.022	$\binom{-.099}{.995}$	1
6	2.616	.777	-2.866	$\binom{.287}{-.958}$	1
7	3.850	-3.347	-2.710	$\binom{-.099}{.995}$	1
8	3.447	.704	-2.558	$\binom{.287}{-.958}$	1

The problem considered is

$$P = \{\binom{x_1}{x_2}: -.099x_1 + .995x_2 + .995 \geqq 0,$$
$$.287x_1 - .958x_2 - 2.873 \geqq 0,\ x_1 \geqq 0,\ x_2 \geqq 0\}.$$



λ = 1.5



λ = .75

FIGURE 24

From the proof of Theorem 3.4 we have

$$d^2(x^{n+1}, P) \leqq d^2(x^n, P) - \frac{\lambda(2-\lambda)[\Sigma\lambda_i(x^n)(a_i'x^n + b_i)]^2}{(\Sigma\lambda_i(x^n)a_i)'(\Sigma\lambda_i(x^n)a_i)}$$

Thus to get the best set of λ_i's we may solve the following optimization problem

$$\max \frac{(\Sigma\lambda_i(-a_i'x^n - b_i))^2}{\|\Sigma\lambda_i a_i\|^2} \equiv k$$

Suppose λ_i^* solves the above, then $\dfrac{\lambda_i^*}{\|\Sigma\lambda_i^* a_i\|}$ also solves the above.

Hence an equivalent problem is

$$\max (\Sigma\lambda_i(-a_i'x^n - b_i))^2$$

$$\text{s.t. } \|\Sigma\lambda_i a_i\|^2 = 1.$$

In our specific problem

$$\max \lambda_1 + 2\lambda_2$$

$$\text{s.t. } \lambda_1^2 + \lambda_2^2 = 1$$

or $\max \sqrt{1 - \lambda_2^2} + 2\lambda_2$.

This gives $\lambda_1^* = \frac{1}{\sqrt{5}}$ and $\lambda_2^* = \frac{2}{\sqrt{5}}$.

This set of λ_is gives

$x^2 = \binom{1}{2}$ which is feasible with $k = 5$.

Had we specified $\lambda_1 = \lambda_2$ i.e., $\lambda_i = \frac{1}{\sqrt{2}}$

$x^2 = \begin{pmatrix} \frac{3}{2} \\ \frac{3}{2} \end{pmatrix}$ which is not feasible and $k = 4.5$.

Thus our prescription does not lead to the best set of λ_is, it is however a good heuristic and easy to implement.

(3) $\underline{\lambda_j(x)}$. In Generalized Merzlyakov method, the weights $\lambda_j(x)$ are determined by the point alone and are not fixed for a subcavity. The following appears to be intuitively a good set of weights since the weightage is dependent on the magnitude of violation of a constraint at the specific point

$$\lambda_j(x) = \left(\frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right)$$

Computational Results

The algorithm described in the previous section was coded primarily to come up with a favorable set of values for the arbitrary parameters λ in Phase I and λ and the weights $\lambda_j(x)$ for Phase II. It may be stated at the outset that selection of these parameters is an art rather than a science and it would be hard to specify the best set of parameters for the general case.

The test problem considered was the problem on "Lease-Buy Planning Decisions" on pages 93 to 114 of Salkin and Saha's "Studies in Linear Programming." The problem has 17 constaints and 20 variables. Together with dual and non-negativity, set M thus had 74 constraints. Set C consisted of the constraint obtained by using strong duality. This problem has a unique solution. The constraints of the problem are reproduced in Appendix.

Test results corroborate the general guidelines for selection of these parameters indicated in the previous section. The distance from a point of P where P = M ∩ C, was calculated at each iteration. The number of iterations required to reduce the distance from P by a certain amount was used as the criterion of improvement. Two

different starting points were selected. Results of computations are summarized below:

Phase I

λ. The relaxation parameter λ was varied in the interval (0,2). For the first starting point, the number of iterations required to reduce the distance d from P from d = 74 to d = 60, was noted, and for the second point the number iterations required to reduce the distance from d = 135 to d = 132 was noted. These results are summarized in Tables 4 and 5 respectively. The results are in line with our expectation based on intuition that a higher λ has the effect of accelerating convergence.

Phase II

The Phase II relaxation parameter was varied in the interval (0,2) and the following alternatives were tested for the weights $\lambda_j(x)$:

(a) $\lambda_j(x) = \dfrac{1}{\text{\# of violated constraints at } x}$

(b) $\lambda_j(x) = \dfrac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}$

(c) $\lambda_j(x) = \left(\dfrac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right)^2$

(d) $\lambda_j(x) = \left(\dfrac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right)^3$

(e) $\lambda_j(x) = \left(\dfrac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right)^4$

(f) If $\left(\dfrac{\text{violation of constraint } j \text{ at } x}{\text{total of violation at } x}\right) \geq .3$ then

$\lambda_j(x) = 10*\left(\dfrac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right)$

if $.2 \leq \left(\dfrac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right) < .3$ then

$\lambda_j(x) = 5*\left(\dfrac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right)$

TABLE 4
Initial d = 74

λ	Final d	No of iterations
.25	63.757	> 20,000
.5	60.0	12,955
.75	59.999	7,339
1.0	60.0	4,866
1.25	59.997	3,469
1.5	59.997	2,565
1.75	59.996	1,927
1.95	59.997	1,525

TABLE 5
Initial d = 135.

λ	Final d	No. of iterations
.25	133.113	> 20,000
.5	132.0	12,698
.75	132.0	6,692
1.0	132.0	4,208
1.25	132.0	2,934
1.5	131.999	2,137
1.75	131.999	1,587
1.95	131.999	1,263

$$\text{if } .1 \leq \left(\frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right) < .2 \text{ then}$$

$$\lambda_j(x) = 3.3*\left(\frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right)$$

$$\text{if } \left(\frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right) < .1 \text{ then}$$

$$\lambda_j(x) = \left(\frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right)$$

(g) $$\text{If } \left(\frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right) \geq .3 \text{ then}$$

$$\lambda_j(x) = 100*\left(\frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right)$$

$$\text{if } .2 \leq \left(\frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right) < .3 \text{ then}$$

$$\lambda_j(x) = 10*\left(\frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right)$$

$$\text{if } .1 \leq \left(\frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right) < .2 \text{ then}$$

$$\lambda_j(x) = 3.3*\left(\frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right)$$

$$\text{if } \left(\frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right) < .1 \text{ then}$$

$$\lambda_j(x) = \left(\frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}\right)$$

Results of alternative (a) are given in Tables 6 and 7 for the same two points used in Phase I.

From Tables 6 and 7 it would appear that again λ of around 2 gives best convergence, but the counterexample in the previous section shows that this is not always the best policy with this set of weights.

Results of alternative (b) are given in Table 8.

TABLE 6

(a) $\lambda_j(x) = \frac{1}{\text{\# of violated constraints at } x}$.

Initial d = 74

λ	Final d	No. of iterations
.25	--	time limit exceeded
.5	--	time limit exceeded
.75	60.0	14,839
1.0	60.0	8,028
1.25	59.998	4,702
1.5	59.998	2,861
1.75	60.0	1,424
1.95	59.857	850

TABLE 7

(a) $\lambda_j(x) = \frac{1}{\text{\# of violated constraints at } x}$.

Initial d = 135

λ	Final d	No. of iterations
.25	--	time limit exceeded
.5	133.45	>20,000
.75	132.0	15,249
1.0	131.999	7,568
1.25	131.999	4,135
1.5	131.999	2,435
1.75	131.998	1,212
1.95	132.0	713

TABLE 8

(b) $\lambda_j(x) = \frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}$

Intiial d = 74

λ	Final d	No. of iterations
.25	59.999	1,253
.5	59.964	1,396
.75	59.997	1,505
1.0	60.0	1,084
1.25	59.999	1,254
1.5	59.976	1,061
1.75	59.997	863
1.95	59.998	894

Alternative (b) appears comparatively insensitive to the relaxation parameter λ. Results of alternatives (b), (c), (d) and (e) are tabulated below:

TABLE 9

Alternative (b): $\lambda_j(x) = \left(\frac{\text{violation of constraint j at x}}{\text{total violation at x}}\right)$

Alternative (c): $\lambda_j(x) = \left(\frac{\text{violation of constraint j at x}}{\text{total violation at x}}\right)^2$

Alternative (d): $\lambda_j(x) = \left(\frac{\text{violation of constraint j at x}}{\text{total violation at x}}\right)^3$

Alternative (e): $\lambda_j(x) = \left(\frac{\text{violation of constraint j at x}}{\text{total violation at x}}\right)^4$

Initial d = 74 λ = .25

Iteration no.	d			
	Alternative (b)	Alternative (c)	Alternative (d)	Alternative (e)
1	73.881	73.883	73.885	73.886
2	73.868	73.874	73.878	73.880
3	73.826	73.836	73.859	73.870
4	73.812	73.821	73.851	73.860
5	73.801	73.754	73.824	73.853
6	73.792	73.747	73.819	73.841
7	73.786	73.737	73.813	73.836
8	73.749	73.722	73.807	73.817
9	73.741	73.718	73.803	73.813
10	73.737	73.681	73.673	73.809

Prima-facie alternative (c) and (d) looked comparable to alternative (b). So the number of iterations required to reduce d to 60.0 was determined and is given in Table 10.

From Table 10 there appears no significant difference between alternative (b), (c), and (d).

Results of alternatives (b), (f), and (g) are shown in Table 11. Since no significant difference in performance was observed, (f) and (g) were not pursued any further.

Results of above study show that the following constitute a favorable set of parameters:

<u>Phase</u> I. $\lambda = 2$

<u>Phase</u> II. $\lambda = 2$

$$\lambda_j(x) = \frac{\text{violation of constraint } j \text{ at } x}{\text{total violation at } x}$$

However as indicated earlier, there are simple counterexamples showing alternative values of the parameters which give better convergence. Hence the above prescriptions can at best be considered good heuristics.

For Phase I of above study, we moved in the direction of most violated constraint of M. Investigations were also conducted combining Phase I and II. Instead of moving in the direction of the most violated constraint of M, we used Generalized Merzlyakov method for relaxation with respect to M. The results were very heartening and are shown in Table 12. In this table a comparison has been made with alternative (b) of Phase II.

TABLE 10

Initial d = 74

	λ	Final d	# of iterations
Alternative (b)	1.95	59.998	894
Alternative (c)	1.95	59.989	956
Alternative (d)	1.95	59.982	1,048

TABLE 11

Initial d = 74   λ = .25

Iteration number	d		
	Alternative (b)	Alternative (f)	Alternative (g)
1	73.881	73.883	73.883
2	73.868	73.875	73.875
3	73.826	73.861	73.861
4	73.812	73.844	73.844
5	73.801	73.836	73.836
6	73.792	73.831	73.831
7	73.786	73.809	73.806
8	73.749	73.803	73.800
9	73.741	73.797	73.794
10	73.737	73.794	73.791

TABLE 12
Initial d = 74 Final d = 60

λ	# of iterations	
	Phase I & II combined	Alternative (b) of Phase II
.1	1882	1686
.25	740	1253
.3	656	1590
.4	483	1241
.45	381	1427
.46	352	1354
.47	322	1273

Concluding Remarks

The simplex method of solving linear programs has achieved its present popularity in a large measure due to the advances during the last three decades on extending the effectiveness of the basic algorithm introduced by Dantzig in 1951. There are however some useful large-scale linear programs beyond the capability of the simplex method which are not being formulated and solved at present due perhaps to their size. It is for such problems that iterative techniques seem to be a viable alternative. We have surveyed two such techniques -- relaxation methods and subgradient methods -- in considerable detail. It has also been shown that most of the recent subgradient methods are mathematically subsumed by the Generalized Merzlyakov method. The latter along with relaxation methods in general are computationally far simpler. In this dissertation another algorithm using relaxation method has been presented which possesses a linear rate of convergence. The new algorithm has been coded and an attempt made to find for it a favorable set of parameters. A variant of the algorithm has been seen experimentally to have considerable merit over the present relaxation methods. Convergence properties of this algorithm are still not attractive enough to be of practical value. However, it is hoped that these could be improved upon and an algorithm found that can economically solve large scale unstructured linear programs.

APPENDIX

The Test Problem

$$\begin{aligned}\text{min. } & 1516x_1 + 1238x_2 + 963x_3 + 750x_4 + 450x_5 + 1200x_6 + 1430x_7 \\ & + 1300x_8 + 1260x_9 + 1300x_{10} + 870x_{11} + 500x_{12} + 1400x_{13} \\ & + 1360x_{14} + 2160x_{15} + 2325x_{16} + 1095x_{17} + 2720x_{18} \\ & + 1440x_{19} + 500x_{20}\end{aligned}$$

subject to

$$x_{16} + x_{17} + x_{18} + x_{19} \geqq 54$$

$$x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} \geqq 54$$

$$x_1 + x_6 + x_7 + x_{14} + x_{15} + x_{16} + x_{18} + x_{19} \geqq 58$$

$$x_1 + x_2 + x_6 + x_7 + x_8 + x_9 + x_{13} + x_{14} + x_{15} + x_{16} + x_{18} \geqq 60$$

$$x_1 + x_2 + x_3 + x_6 + x_7 + x_8 + x_{10} + x_{13} + x_{14} + x_{15} + x_{16} + x_{18} \geqq 65$$

$$x_1 + x_2 + x_3 + x_4 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{13} + x_{15} + x_{16} + x_{18} \geqq 65$$

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_7 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{15} + x_{18} + x_{20} \geqq 70$$

$$x_{16} + x_{17} = 12$$

$$x_{18} + x_{19} = 17$$

$$x_{13} \leqq 10$$

$$x_{14} + x_{15} \leqq 15$$

$$-x_{16} + x_{20} \leqq 0$$

$$x_1 + x_2 + x_6 + x_7 + x_8 + x_9 - x_{14} - x_{15} + x_{17} + x_{19} \leqq 36$$

$$x_3 + x_4 + x_{10} + x_{11} - x_{13} \leqq 40$$

$$x_5 + x_6 + x_8 + x_{12} + x_{14} + x_{16} - x_{20} \leqq 45$$

$$x_{10} + x_{11} + x_{13} + x_{18} \leqq 50$$

$$x_7 + x_9 + x_{12} + x_{17} + x_{20} \leqq 50$$

$$x_i \geqq 0,\ i = 1, \ldots, 20$$

Note: Problem abstracted from Case Study on "Lease-Buy Planning Decisions" in Salkin, H. M., and J. Saha Studies in Linear Programming, 1975, pp. 93-114.

BIBLIOGRAPHY

1. Agmon, S., "The Relaxation Method for Linear Inequalities," Canadian Journal of Mathematics, 6, No. 3, 1954, pp. 382-392.

2. Beale, E. M. L., "The Current Algorithmic Scope of Mathematical Programming Systems," Mathematical Programming Study 4, 1975, pp. 1-11.

3. Bland, R. G., "New Finite Pivoting Rules for the Simplex Method.," Mathematics of Operations Research, 2, 1977, pp. 103-107.

4. Busacker, R. G. and T. L. Saaty, Finite Graphs and Networks, McGraw-Hill, New York, 1965.

5. Camerini, P. M., L. Fratta, and F. Maffioli, "On Improving Relaxation Methods by Modified Gradient Techniques," Mathematical Programming Study 3, 1975, pp. 26-34.

6. Dantzig, G. B., "Maximization of a Linear Function of Variables Subject to Linear Inequalities," in T. C. Koopman, ed. Activity Analysis of Production and Allocation, Wiley, New York, 1951, Ch. XXI.

7. Dantzig, G. B., Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey, 1963, pp. 84-85.

8. Dantzig, G. B., R. W. Cottle, B. C. Eaves, F. S. Hillier, A. S. Manne, G. H. Golub, D. J. Wilde and R. B. Wilson, "On the Need for a System Optimization Laboratory," in T. C. Hu and S. M. Robinson, eds. Mathematical Programming, Academic Press, New York, 1973, p. 2.

9. Dantzig, G. B. and R. Van Slyke, "Generalized Upper Bounding Techniques," Journal of Computer System Science, 1968, pp. 213-226.

10. Demjanov, V. F., "Algorithms for some minimax problems," Journal of Computer and System Sciences 2 (1968), pp. 342-380.

11. Eaves, B. C., "Piecewise Linear Retractions by Reflextion," Linear Algebra and its Applications, 7, 1973, pp. 93-98.

12. Eremin, I. I., "An Iterative Method for Cebysev Approximations of Incompatible Systems of Linear Inequalities," Soviet Mathematical Doklady, 1961, pp. 821-824.

13. Fejer, L., "Ueber die Lage der Nulstellen von Polynomen die aus Minimum forderungen genisser Arten entspringen," Mathematical Annalen, 85, 1922, pp. 41-48.

14. Forrest, J. J. H. and J. A. Tomlin, "Updating Triangular Factors of the Basis in the Product Form Simplex Method," Mathematical Programming 2, 1972, pp. 263-278.

15. Goffin, J. L., On the Finite Convergence of the Relaxation Method for Solving Systems of Inequalities, Dissertation University of California, Berkeley, 1971.

16. Goffin, J. L., "On Convergence Rates of Subgradient Optimization Methods," Working Paper No. 34, McGill University, 1976.

17. Harris, P. M. J., "Pivot Selection Methods of the Devex LP Code," Mathematical Programming Study 4, 1975, pp. 30-57.

18. Held, M. and R. M. Karp, "The Travelling-Salesman Problem and Minimum Spanning Trees: Part II," Mathematical Programming, 1, 1971, pp. 6-25.

19. Held, M., P. Wolfe, and H. Crowder, "Validation of Subgradient Optimization," Mathematical Programming, 6, 1974, pp. 62-88.

20. Hellerman, E. and D. C. Rarick, "The Partitioned Preassigned Pivot Procedure (P^4)," in D. J. Rose and R. A. Willoughby eds., Sparse Matrices and Their Applications, Plenum Press, New York, 1972.

21. Hoffman, A. J., "On Approximate Solutions of Systems of Linear Inequalities," Journal of Research of the National Bureau of Standards, 48, 4, 1952, pp. 263-265.

22. Kalan, J. E., "Aspects of Large-Scale, In-Core Linear Programming," Proceedings of ACM Annual Conference, Chicago, August, 1971.

23. Koehler, G. J., "A Case for Relaxation Methods in Large-Scale Linear Programming," Large Scale Systems Theory and Applications, Proceedings of the IFAC Symposium, June 1976, Udine, Italy.

24. Koehler, G. J. and G. S. Kumar, "A Look at Finding Points of a Convex Polyhedron Using Relaxation and Subgradient Procedures," to appear in Operations Research.

25. Koehler, G. J., A. B. Whinston and G. P. Wright, Optimization over Leontief Substitution Systems, North-Holland Publishing Company, Amsterdam, 1975.

26. Kohler, D. A., Projections of Convex Polyhedral Sets, Dissertation, University of California, Berkeley, 1967.

27. Lasdon, L. S., Optimization Theory for Large Systems, The MacMillan Company, New York, 1970.

28. Lemarchel, C., "An Extension of Davidon Methods to Non-differentiable Problems," Mathematical Programming Study, 3, 1975, pp. 95-109.

29. Markovitz, H. M., "The Elimination Form of Inverse and its Application to Linear Programming," Management Science, 3, 1957, pp. 255-269.

30. Merzlyakov, Y. I., "On a Relaxation Method of Solving Systems of Linear Inequalities," USSR Computational Mathematics and Mathematical Physics, 2 (3), 1963, pp. 504-510.

31. Motzkin, T. S. and I. J. Schoenberg, "The Relaxation Method for Linear Inequalities," Canadian Journal of Mathematics, 6, No. 3, 1954, pp. 393-404.

32. Oettli, W., "An Iterative Method, Having Linear Rate of Convergence, for Solving a Pair of Dual Linear Programs," Mathematical Programming, 3, 1972, pp. 302-311.

33. Oettli, W., "Symmetric Duality and a Convergent Subgradient Method for Discrete, Linear, Constrained Approximation Problems with Arbitrary Norms Appearing in the Objective Function and in the Constraints," Journal of Approximation Theory, 14, 1, 1975, pp. 43-50.

34. Polyak, B. T., "A General Method of Solving Extremum Problems," Soviet Mathematical Doklady, 1967, pp. 593-597.

35. Polyak, B. T., "Minimization of Unsmooth Functions," USSR Computational Mathematics and Mathematical Physics, 1969, pp. 14-29.

36. Shor, N. Z., On the Structure of Algorithms for the Numerical Solution of Optimal Planning and Design Problems, Dissertation, Cybernetics Institute An, Kiev, 1974.

37. Wolfe, P., "A Method of Conjugate Subgradients for Minimizing Non-differentiable Functions," Proceedings Twelfth Annual Allerton Conference on Circuit and System Theory, Chicago, October 2-4, 1974, pp. 8-15.

38. Wolfe, P., "A Method of Conjugate Subgradients for Minimizing Non-differentiable Functions," Mathematical Programming Study 3, 1975, pp. 145-173.

BIOGRAPHICAL SKETCH

Gollakota Surya Kumar was born in Kakinada, India, on April 5, 1939. After secondary schooling at Modern School, New Delhi, India he joined St. Stephens College, Delhi University in BSc (Physics Honours) in July 1956. After two years in Delhi University he was one of 24 students selected in an all-India competition to pursue a four-year course (1958 to 1962) in mechanical engineering at Indian Railways School of Mechanical and Electrical Engineering at Jamalpur, India. During this period he passed Parts I, II and III of Institution of Mechanical Engineers (London). He became a Graduate of Institution of Mechanical Engineers (London) in July 1962 and was Elected a Member of the British Institution in November 1969.

From June 1964 to March 1970 he worked as Assistant Plant Manager and Plant Manager of three major plants of Indian Railways employing a total of over 10,000 persons. From March 1970 to August 1972 he worked as Assistant Director, Research, Design and Standards Organization of Indian Railways.

In August 1972 he came to the United States and joined the MBA Program at Indiana University, Bloomington, Indiana. From January 1976 he has been pursuing doctoral studies in business administration with a major in management at the University of

Florida. He has been teaching senior level courses in managerial operations analysis over the last two years. He has recently accepted a position as an assistant professor in the College of Business Administration, Northern Illinois University, Dekalb, Illinois. He was General Editor of his school weekly "Sandesh" and has several professional papers to his credit in India and USA. His hobbies include badminton, basketball and photography. He presented a one-man exhibition of photographs in Jamalpur entitled, "Four Years in Jamlpur."

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Gary J. Koehler, Chairman
Associate Professor of Management

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Antal Majthay
Associate Professor of Management

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Jorge Martinez
Associate Professor of Mathematics

This dissertation was submitted to the Graduate Faculty of the Department of Management in the College of Business Administration and to the Graduate Council, and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August, 1979

Dean, Graduate School